DEVELOPMENT APPROACHES COUPLED WITH VERIFICATION AND VALIDATION METHODOLOGIES FOR AGENT-BASED MISSION-LEVEL ANALYTICAL COMBAT SIMULATIONS

DISSERTATION

Lance E. Champagne, Major, USAF

AFIT/DS/ENS/03-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DEVELOPMENT APPROACHES COUPLED WITH VERIFICATION AND VALIDATION METHODOLOGIES FOR AGENT-BASED MISSION-LEVEL ANALYTICAL COMBAT SIMULATIONS

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Lance E. Champagne, B.S., B.S.E., M.S.

Major, USAF

6 November 2003

AFIT/DS/ENS/03-02

DEVELOPMENT APPROACHES COUPLED WITH VERIFICATION AND VALIDATION METHODOLOGIES FOR AGENT-BASED MISSION-LEVEL ANALYTICAL COMBAT SIMULATIONS

Lance E. Champagne, B.S., B.S.E, M.S.
Major, USAF

Approved:

	Date
//Signed// Dr. Raymond R. Hill (Chairman)	4 March 2004
//Signed// Dr. Kenneth W. Bauer (Member)	5 March 2004
//Signed// Dr. James T. Moore (Member)	5 March 2004
//Signed// Dr. Mark Oxley (Member)	5 March 2004
//Signed// Dr Richard Raines (Dean's Representative)	8 March 2004

Accepted:

//Signed// 12 March 2004
Dr. Robert A. Calico, Jr. Date
Dean, Graduate School of Engineering and Management

AFIT/DS/ENS/03-02

Abstract

This research investigated the applicability of agent-based combat simulations to real-world combat operations. An agent-based simulation of the Allied offensive search for German U-Boats in the Bay of Biscay during World War II was constructed, extending the state-of-the-art in agent-based combat simulations, bridging the gap between the current level of agent-like combat simulations and the concept of agent-based simulations found in the broader literature. The proposed simulation advances agent-based combat simulations to "validateable" mission-level military operations.

Simulation validation is a complex task with numerous, diverse techniques available and levels of validation differing significantly among simulations and applications. This research presents a verification and validation taxonomy based on face validity, empirical validity, and theoretical validity, extending the verification and validation knowledge-base to include techniques specific to agent-based models. The verification and validation techniques are demonstrated in a Bay of Biscay case study.

Validating combat operations pose particular problems due to the infrequency of real-world occurrences to serve as simulation validation cases; often just a single validation comparison can be made. This means comparisons to the underlying stochastic process are not possible without significant loss of statistical confidence. This research also presents a statistical validation methodology based on re-sampling historical outcomes, which when coupled with the traditional nonparametric sign test, allows comparison between a simulation and historic operation providing an improved validation indicator beyond the single pass or fail test.

Acknowledgements

I would like to express my sincere appreciation to my faculty advisor, Professors Ray Hill, Jr., for guidance, support and leadership. My committee members, Professor Kenneth Bauer, Professor Mark Oxley, and Professor James Moore have been generous with their time and support, and they have my most profound thanks. Each of these individuals demonstrates the highest degree of professionalism, both in the academic and military spheres. All the faculty members with whom I have had contact have been exemplary. In particular I wish to thank Professors Miller and Kharoufeh.

I would be remiss if I failed to acknowledge the professionalism, diligence, and acuity of the departmental and library staffs. I am grateful in particular to Kathy McBride, Jan Farr, Joel Coons, Rhonda Sewell, and Robin Hays. I am especially grateful to my fellow Ph.D. students, past and present. Particularly, Maj Trevor Laine provided a much-appreciated sounding board as an officemate and friend. A special thank you goes to Lt Col Tom Tighe for his military expertise and big-picture input.

I also wish to express my gratitude to Dr. Mike Young of the Air Force Research Laboratory for his time and support. The USMC Project Albert team was especially helpful, particularly Dr. Gary Horne and Ms. Sarah Johnson. Dr. Brian McCue was a source of unparalleled expertise, and I thank him for his valuable aid. I also need to acknowledge the support and expertise given by members of the Defense Modeling and Simulation Office, especially Colonel Eileen Bjorkman and Dr. Sue Numrich.

While I have learned much from formal schools, I have learned much more from peers, superiors and, especially, subordinates under challenging circumstances. I owe them much of my success.

Finally, I am indebted to my family and friends. Though last in this list, they deserve the greatest level of praise and honor for their encouragement, sacrifices,

compromises, and understanding. Without their unflagging support and encouragement I would have floundered long ago. I am grateful to all. They have my undying gratitude and love.

Lance E. Champagne

Table of Contents

Page

Abstract ... iv

Acknowledgements ... v

List of Figures ... xi

List of Tables ... xiii

I. Introduction ... 1

- 1.1 Introduction ... 1
- 1.2 DoD Simulations ... 4
  - 1.2.1 Constructive Simulation Classification ... 4
  - 1.2.2 Agent-Based Simulation ... 8
- 1.3 Research Goal ... 11
  - 1.3.1 Establishing the Background and Supporting Work ... 11
  - 1.3.2 Extend Agent-Based Combat Simulations to the Mission-Level ... 12
  - 1.3.3 Develop Validation Methods for Agent-Based Combat Simulations ... 12
  - 1.3.4 Demonstration of Methods via Known Use-Case ... 13
- 1.4 Contributions of this Research ... 14
- 1.5 Sequence of Presentation ... 15

II. Agent-Based Simulation ... 16

- 2.1 Agent Defined ... 17
  - 2.1.1 Differentiating Between Discrete-Event, Object-Oriented, and Agent-Based Simulations ... 19
  - 2.1.2 Differentiating Further Between “Agents” and “Objects” ... 24
  - 2.1.3 Types of Agent Behavior ... 25
  - 2.1.4 Agent-Based Programming Defined ... 26
  - 2.1.5 Properties of Agent-Based Systems ... 28
- 2.2 Types of Agent Systems and Uses ... 33
- 2.3 Agent-based Combat Simulation ... 34
- 2.4 Adaptation ... 37
  - 2.4.1 Types of agent adaptation ... 38
    - 2.4.1.1 Learning ... 38

Page

2.4.1.2 Artificial Neural Networks 41
2.4.1.3 Genetic Algorithms 41
2.4.2 A New Approach to Agent Adaptation 42
2.5 Conclusion 43

III. Simulation Validation and Verification Methodology and Taxonomy 44

3.1 Definitions 46
3.2 Taxonomy 49
3.2.1 Verification Classifications 50
3.2.1.1 Software Engineering Practices 51
3.2.1.2 Static Verification 53
3.2.1.3 Dynamic Verification 54
3.2.2 Validation classifications 57
3.2.2.1 Face Validity 58
3.2.2.2 Empirical Validity 60
3.2.2.3 Theoretical Validity 62
3.3 V&V Methodology 63
3.4 Does V&V Ultimately Matter? 68

IV. Bay of Biscay Agent-Based Simulation 69

4.1 The Historical Operation 69
4.2 Model Description 70
4.2.1 Assumptions 71
4.2.1.1 Environment 71
4.2.1.2 U-Boat Assumptions 73
4.2.1.3 Aircraft Assumptions 75
4.2.2 Conceptual Models 77
4.2.3 Conceptual Model Validation 79
4.2.3.1 Validation against previously validated models. 79
4.2.3.2 Prototyping and Subject Matter Experts. 81
4.2.3.3 Preliminary Output Analysis 81
4.2.4 Conceptual Model Implementation 81
4.2.4.1 Agent Decisions and Movement 82
4.2.4.2 Aircraft Search 89
4.2.4.3 Agent Strategy and Adaptation 96
4.2.4.4 Other Agent-Based Issues 99
4.2.4.5 Model Verification 99

Page

4.2.5 Agent Adaptation 104
4.2.5.1 Aircraft Adaptation 105
4.2.5.2 U-Boat Adaptation 108
4.2.6 Simulation Output Format 110
4.2.6.1 Iteration Total 111
4.2.6.2 Mean Total Value 111
4.2.6.3 Iteration Mean Monthly Value 111
4.2.6.4 Overall Mean Monthly Value 111
4.3 Analysis Objectives 112
4.3.1 MOEs 113
4.3.2 Validation Scenarios 113
4.3.3 Validation Criteria 118
4.4 Model Output and Validation 119
4.4.1 Gauging the Allied Level of Effort 120
4.4.2 Validation of Scenario 1 Results 124
4.4.3 Validation of Scenario 2 Results 126
4.4.4 Validation of Emergent Behavior 129
4.4.5 Validation Conclusions 131
4.5 Extensions to Modern Problems 132
4.5.1 Scenario Fundamentals 132
4.5.2 Possible Modern Applications 133
4.5.3 Summary 136
4.6 Conclusion 136

V. New Statistical Approach to Validating Agent-Based Combat Simulations 138

5.1 Motivation for a New Validation Test 138
5.2 Methodology for Comparison of Historic versus Simulated Data 147
5.2.1 Bootstrap 148
5.2.2 Sign Test 151
5.3 Bay of Biscay Agent-Based Simulation Results 152
5.3.1 Sortie Hours 152
5.3.2 Scenario 1 MOEs 160
5.3.3 Scenario 2 MOEs 167
5.3.4 Validation Conclusions 174
5.4 Contributions 175

VI. Contributions and Avenues for Future Research 176

6.1 Contributions 176
6.1.1 Establishing the Background and Supporting Work 176

Page

6.1.2 Extend Agent-Based Combat Simulations to the Mission-Level 177
6.1.3 Develop Validation Methods for Agent-Based Combat Simulations 178
6.1.4 Demonstration of Methods via Known Use-Case 179
6.2 Future Research 179
6.2.1 Additional Agent Behaviors 179
6.2.2 Modern Scenario Extensions 180

Appendix A. Bootstrap Results for Simulation MOEs 181

A.1 Scenario 1 Sortie Hours 181
A.2 Scenario 2 Sortie Hours 192
A.3 Scenario 1 U-Boat Sightings 203
A.4 Scenario 1 U-Boat Kills 214
A.5 Scenario 2 U-Boat Sightings 225
A.6 Scenario 2 U-Boat Kills 236

Appendix B. Model Implementation 247

B.1 Aircraft Agent Algorithms 248
B.2 U-Boat Agent Algorithms 253
B.3 Simulation Environment 256

References 259

List of Figures

Page

Figure 1.1 Modeling pyramid with representative models..5

Figure 3.1 Verification and Validation Taxonomy...50

Figure 3.2 Modeling and Simulation Process...64

Figure 3.3 Generalized Modeling Process with Feedback..65

Figure 3.4 Simplified Modeling Process (Sargent, 1996a)...67

Figure 4.1 U-Boat Flow, Conceptual Model..78

Figure 4.2 Aircraft Flow, Conceptual Model..79

Figure 4.3 Possible Agent Moves...83

Figure 4.4 Search Zone in the Bay of Biscay...89

Figure 4.5 Complete Aircraft Search Grid..90

Figure 4.6 Modified Barrier Search Pattern..91

Figure 4.7 Aircraft Agent Search..92

Figure 4.8 Parallel Search Pattern...93

Figure 4.9 Creeping Line Search Pattern..94

Figure 4.10 Square Search Pattern..94

Figure 4.11 Sector Search Pattern...95

Figure 4.12 Generic Aircraft Agent Scheduling Process for Day versus Night Missions..107

Figure 4.13 Variance Reduction in Pre-Production Model, Scenario 1........................117

Figure 4.14 Variance Reduction in Pre-Production Model, Scenario 2........................117

Figure 4.15 Total Sortie Hours Flown, Combined Scenarios......................................123

Page

Figure 4.16 Comparisons of Simulated versus Historical MOE Values, Scenario 1...... 126

Figure 4.17 Comparisons of Simulated versus Historical MOE Values, Scenario 2...... 128

Figure 5.1 Comparisons of Mean Monthly Sortie Hours, Historic vs. Simulated Scenario 1... 140

Figure 5.2 Comparisons of Mean Monthly Sortie Hours, Historic vs. Simulated Scenario 2... 141

Figure 5.3 Comparisons of Mean Monthly U-Boat Sightings, Historic vs. Simulated Scenario 1 143

Figure 5.4 Comparisons of Mean Monthly U-Boat Kills, Historic vs. Simulated Scenario 1... 144

Figure 5.5 Comparisons of Mean Monthly U-Boat Sightings, Historic vs. Simulated Scenario 2 145

Figure 5.6 Comparisons of Mean Monthly U-Boat Kills, Historic vs. Simulated Scenario 2... 146

Figure 5.7 Methodology for Comparisons of a Single-Sampled Real-World Process to Simulated Results 147

Figure B.1 Simulation Class Inheritance Diagram 247

Figure B.2 Bomber Agent Run Method Algorithm 249

Figure B.3 Bomber Agent Update Method Algorithm 250

Figure B.4 Bomber Agent U-Boat Detection Algorithm 252

Figure B.5 U-Boat Agent Run Method Algorithm 254

Figure B.6 U-Boat Agent Update Method Algorithm 255

Figure B.7 Field getMinUpdate Method Algorithm Used to Advance the Simulation Clock and Control Agent Timing 257

List of Tables

Page

Table 2.1 Four Categories of AI Study 25

Table 4.1 U-Boat Reinforcements for Validation Scenarios [McCue, 1990] 115

Table 4.2 Historical MOE values for Scenario 1 [McCue, 1990] 119

Table 4.3 Historical MOE values for Scenario 2 [McCue, 1990] 119

Table 4.4 Simulated Aircraft Sortie Hours for Scenario 1 121

Table 4.5 Simulated Aircraft Sortie Hours for Scenario 2 121

Table 4.6 Total Sortie Hours, Simulated versus Actual 122

Table 4.7 Simulated U-Boat Sightings for Scenario 1 124

Table 4.8 Simulated U-Boat Kills for Scenario 1 125

Table 4.9 Combined MOEs for Scenario 1, Simulated versus Actual 125

Table 4.10 Simulated U-Boat Sightings for Scenario 2 127

Table 4.11 Simulated U-Boat Kills for Scenario 2 127

Table 4.12 Combined MOEs for Scenario 2, Simulated versus Actual 128

Table 4.13 U-Boat Inter-arrival Statistics and Index of Dispersion 130

Table 4.14 Index of Dispersion for U-Boat Inter-arrival Times 130

Table 5.1 Autocorrelation of Historic MOE Values 149

Table 5.2 Bootstrap Sortie Hours – Scenario 1 154

Table 5.3 Sign Test Calculations – Sortie Hours, Scenario 1 155

Table 5.4 Summary of 20 Bootstrap Experiments for Scenario 1 Sortie Hours 156

Table 5.5 Bootstrap Sortie Hours – Scenario 2 157

Table 5.6 Sign Test Calculations – Sortie Hours, Scenario 2 158

Page

Table 5.7 Summary of 20 Bootstrap Experiments for Scenario 2 Sortie Hours............ 159

Table 5.8 Bootstrap U-Boat Sightings – Scenario 1 161

Table 5.9 Sign Test Calculations – U-Boat Sightings, Scenario 1 162

Table 5.10 Summary of 20 Bootstrap Experiments for Scenario 1 U-Boat Sightings 163

Table 5.11 Bootstrap U-Boat Kills – Scenario 1 164

Table 5.12 Sign Test Calculations – U-Boat Kills, Scenario 1 165

Table 5.13 Summary of 20 Bootstrap Experiments for Scenario 1 U-Boat Kills 166

Table 5.14 Bootstrap U-Boat Sightings – Scenario 2 168

Table 5.15 Sign Test Calculations – U-Boat Sightings, Scenario 2 169

Table 5.16 Summary of 20 Bootstrap Experiments for Scenario 2 U-Boat Sightings 170

Table 5.17 Bootstrap U-Boat Kills – Scenario 2 171

Table 5.18 Sign Test Calculations – U-Boat Kills, Scenario 2 172

Table 5.19 Summary of 20 Bootstrap Experiments for Scenario 2 U-Boat Kills 173

Table A.1 Bootstrap Samples, Replication 1, Scenario 1 Sortie Hours......................... 181

Table A.2 Bootstrap Samples, Replication 2, Scenario 1 Sortie Hours......................... 182

Table A.3 Bootstrap Samples, Replication 3, Scenario 1 Sortie Hours......................... 182

Table A.4 Bootstrap Samples, Replication 4, Scenario 1 Sortie Hours......................... 183

Table A.5 Bootstrap Samples, Replication 5, Scenario 1 Sortie Hours......................... 183

Table A.6 Bootstrap Samples, Replication 6, Scenario 1 Sortie Hours......................... 184

Table A.7 Bootstrap Samples, Replication 7, Scenario 1 Sortie Hours......................... 184

Table A.8 Bootstrap Samples, Replication 8, Scenario 1 Sortie Hours......................... 185

Page

Table A.9 Bootstrap Samples, Replication 9, Scenario 1 Sortie Hours 185

Table A.10 Bootstrap Samples, Replication 10, Scenario 1 Sortie Hours 186

Table A.11 Bootstrap Samples, Replication 11, Scenario 1 Sortie Hours 186

Table A.12 Bootstrap Samples, Replication 12, Scenario 1 Sortie Hours 187

Table A.13 Bootstrap Samples, Replication 13, Scenario 1 Sortie Hours 187

Table A.14 Bootstrap Samples, Replication 14, Scenario 1 Sortie Hours 188

Table A.15 Bootstrap Samples, Replication 15, Scenario 1 Sortie Hours 188

Table A.16 Bootstrap Samples, Replication 16, Scenario 1 Sortie Hours 189

Table A.17 Bootstrap Samples, Replication 17, Scenario 1 Sortie Hours 189

Table A.18 Bootstrap Samples, Replication 18, Scenario 1 Sortie Hours 190

Table A.19 Bootstrap Samples, Replication 19, Scenario 1 Sortie Hours 190

Table A.20 Bootstrap Samples, Replication 20, Scenario 1 Sortie Hours 191

Table A.21 Bootstrap Samples, Replication 1, Scenario 2 Sortie Hours 192

Table A.22 Bootstrap Samples, Replication 2, Scenario 2 Sortie Hours 193

Table A.23 Bootstrap Samples, Replication 3, Scenario 2 Sortie Hours 193

Table A.24 Bootstrap Samples, Replication 4, Scenario 2 Sortie Hours 194

Table A.25 Bootstrap Samples, Replication 5, Scenario 2 Sortie Hours 194

Table A.26 Bootstrap Samples, Replication 6, Scenario 2 Sortie Hours 195

Table A.27 Bootstrap Samples, Replication 7, Scenario 2 Sortie Hours 195

Table A.28 Bootstrap Samples, Replication 8, Scenario 2 Sortie Hours 196

Table A.29 Bootstrap Samples, Replication 9, Scenario 2 Sortie Hours 196

Table A.30 Bootstrap Samples, Replication 10, Scenario 2 Sortie Hours 197

Page

Table A.31 Bootstrap Samples, Replication 11, Scenario 2 Sortie Hours 197

Table A.32 Bootstrap Samples, Replication 12, Scenario 2 Sortie Hours 198

Table A.33 Bootstrap Samples, Replication 13, Scenario 2 Sortie Hours 198

Table A.34 Bootstrap Samples, Replication 14, Scenario 2 Sortie Hours 199

Table A.35 Bootstrap Samples, Replication 15, Scenario 2 Sortie Hours 199

Table A.36 Bootstrap Samples, Replication 16, Scenario 2 Sortie Hours 200

Table A.37 Bootstrap Samples, Replication 17, Scenario 2 Sortie Hours 200

Table A.38 Bootstrap Samples, Replication 18, Scenario 2 Sortie Hours 201

Table A.39 Bootstrap Samples, Replication 19, Scenario 2 Sortie Hours 201

Table A.40 Bootstrap Samples, Replication 20, Scenario 2 Sortie Hours 202

Table A.41 Bootstrap Samples, Replication 1, Scenario 1 U-Boat Sightings 203

Table A.42 Bootstrap Samples, Replication 2, Scenario 1 U-Boat Sightings 204

Table A.43 Bootstrap Samples, Replication 3, Scenario 1 U-Boat Sightings 204

Table A.44 Bootstrap Samples, Replication 4, Scenario 1 U-Boat Sightings 205

Table A.45 Bootstrap Samples, Replication 5, Scenario 1 U-Boat Sightings 205

Table A.46 Bootstrap Samples, Replication 6, Scenario 1 U-Boat Sightings 206

Table A.47 Bootstrap Samples, Replication 7, Scenario 1 U-Boat Sightings 206

Table A.48 Bootstrap Samples, Replication 8, Scenario 1 U-Boat Sightings 207

Table A.49 Bootstrap Samples, Replication 9, Scenario 1 U-Boat Sightings 207

Table A.50 Bootstrap Samples, Replication 10, Scenario 1 U-Boat Sightings 208

Table A.51 Bootstrap Samples, Replication 11, Scenario 1 U-Boat Sightings 208

Page

Table A.52 Bootstrap Samples, Replication 12, Scenario 1 U-Boat Sightings.............209

Table A.53 Bootstrap Samples, Replication 13, Scenario 1 U-Boat Sightings.............209

Table A.54 Bootstrap Samples, Replication 14, Scenario 1 U-Boat Sightings.............210

Table A.55 Bootstrap Samples, Replication 15, Scenario 1 U-Boat Sightings.............210

Table A.56 Bootstrap Samples, Replication 16, Scenario 1 U-Boat Sightings.............211

Table A.57 Bootstrap Samples, Replication 17, Scenario 1 U-Boat Sightings.............211

Table A.58 Bootstrap Samples, Replication 18, Scenario 1 U-Boat Sightings.............212

Table A.59 Bootstrap Samples, Replication 19, Scenario 1 U-Boat Sightings.............212

Table A.60 Bootstrap Samples, Replication 20, Scenario 1 U-Boat Sightings.............213

Table A.61 Bootstrap Samples, Replication 1, Scenario 1 U-Boat Kills.......................214

Table A.62 Bootstrap Samples, Replication 2, Scenario 1 U-Boat Kills.......................215

Table A.63 Bootstrap Samples, Replication 3, Scenario 1 U-Boat Kills.......................215

Table A.64 Bootstrap Samples, Replication 4, Scenario 1 U-Boat Kills.......................216

Table A.65 Bootstrap Samples, Replication 5, Scenario 1 U-Boat Kills.......................216

Table A.66 Bootstrap Samples, Replication 6, Scenario 1 U-Boat Kills.......................217

Table A.67 Bootstrap Samples, Replication 7, Scenario 1 U-Boat Kills.......................217

Table A.68 Bootstrap Samples, Replication 8, Scenario 1 U-Boat Kills.......................218

Table A.69 Bootstrap Samples, Replication 9, Scenario 1 U-Boat Kills.......................218

Table A.70 Bootstrap Samples, Replication 10, Scenario 1 U-Boat Kills.....................219

Table A.71 Bootstrap Samples, Replication 11, Scenario 1 U-Boat Kills.....................219

Table A.72 Bootstrap Samples, Replication 12, Scenario 1 U-Boat Kills.....................220

Table A.73 Bootstrap Samples, Replication 13, Scenario 1 U-Boat Kills.....................220

Page

Table A.74 Bootstrap Samples, Replication 14, Scenario 1 U-Boat Kills 221

Table A.75 Bootstrap Samples, Replication 15, Scenario 1 U-Boat Kills 221

Table A.76 Bootstrap Samples, Replication 16, Scenario 1 U-Boat Kills 222

Table A.77 Bootstrap Samples, Replication 17, Scenario 1 U-Boat Kills 222

Table A.78 Bootstrap Samples, Replication 18, Scenario 1 U-Boat Kills 223

Table A.79 Bootstrap Samples, Replication 19, Scenario 1 U-Boat Kills 223

Table A.80 Bootstrap Samples, Replication 20, Scenario 1 U-Boat Kills 224

Table A.81 Bootstrap Samples, Replication 1, Scenario 2 U-Boat Sightings 225

Table A.82 Bootstrap Samples, Replication 2, Scenario 2 U-Boat Sightings 226

Table A.83 Bootstrap Samples, Replication 3, Scenario 2 U-Boat Sightings 226

Table A.84 Bootstrap Samples, Replication 4, Scenario 2 U-Boat Sightings 227

Table A.84 Bootstrap Samples, Replication 4, Scenario 2 U-Boat Sightings 227

Table A.85 Bootstrap Samples, Replication 5, Scenario 2 U-Boat Sightings 228

Table A.86 Bootstrap Samples, Replication 6, Scenario 2 U-Boat Sightings 228

Table A.87 Bootstrap Samples, Replication 7, Scenario 2 U-Boat Sightings 229

Table A.88 Bootstrap Samples, Replication 8, Scenario 2 U-Boat Sightings 229

Table A.89 Bootstrap Samples, Replication 9, Scenario 2 U-Boat Sightings 230

Table A.90 Bootstrap Samples, Replication 10, Scenario 2 U-Boat Sightings 230

Table A.91 Bootstrap Samples, Replication 11, Scenario 2 U-Boat Sightings 231

Table A.93 Bootstrap Samples, Replication 13, Scenario 2 U-Boat Sightings 231

Table A.94 Bootstrap Samples, Replication 14, Scenario 2 U-Boat Sightings 232

Table A.95 Bootstrap Samples, Replication 15, Scenario 2 U-Boat Sightings 232

Page

Table A.96 Bootstrap Samples, Replication 16, Scenario 2 U-Boat Sightings 233

Table A.97 Bootstrap Samples, Replication 17, Scenario 2 U-Boat Sightings 233

Table A.98 Bootstrap Samples, Replication 18, Scenario 2 U-Boat Sightings 234

Table A.99 Bootstrap Samples, Replication 19, Scenario 2 U-Boat Sightings 234

Table A.100 Bootstrap Samples, Replication 20, Scenario 2 U-Boat Sightings 235

Table A.101 Bootstrap Samples, Replication 1, Scenario 2 U-Boat Kills 236

Table A.102 Bootstrap Samples, Replication 2, Scenario 2 U-Boat Kills 237

Table A.103 Bootstrap Samples, Replication 3, Scenario 2 U-Boat Kills 237

Table A.104 Bootstrap Samples, Replication 4, Scenario 2 U-Boat Kills 238

Table A.105 Bootstrap Samples, Replication 5, Scenario 2 U-Boat Kills 238

Table A.106 Bootstrap Samples, Replication 6, Scenario 2 U-Boat Kills 239

Table A.107 Bootstrap Samples, Replication 7, Scenario 2 U-Boat Kills 239

Table A.108 Bootstrap Samples, Replication 8, Scenario 2 U-Boat Kills 240

Table A.109 Bootstrap Samples, Replication 9, Scenario 2 U-Boat Kills 240

Table A.110 Bootstrap Samples, Replication 10, Scenario 2 U-Boat Kills 241

Table A.111 Bootstrap Samples, Replication 11, Scenario 2 U-Boat Kills 241

Table A.112 Bootstrap Samples, Replication 12, Scenario 2 U-Boat Kills 242

Table A.113 Bootstrap Samples, Replication 13, Scenario 2 U-Boat Kills 242

Table A.114 Bootstrap Samples, Replication 14, Scenario 2 U-Boat Kills 243

Table A.115 Bootstrap Samples, Replication 15, Scenario 2 U-Boat Kills 243

Table A.116 Bootstrap Samples, Replication 16, Scenario 2 U-Boat Kills 244

Table A.117 Bootstrap Samples, Replication 17, Scenario 2 U-Boat Kills 244

Page

Table A.118 Bootstrap Samples, Replication 18, Scenario 2 U-Boat Kills 245

Table A.119 Bootstrap Samples, Replication 19, Scenario 2 U-Boat Kills 245

Table A.120 Bootstrap Samples, Replication 20, Scenario 2 U-Boat Kills 246

DEVELOPMENT APPROACHES COUPLED WITH VERIFICATION AND VALIDATION METHODOLOGIES FOR AGENT-BASED MISSION-LEVEL ANALYTICAL COMBAT SIMULATIONS

I. Introduction

1.1 Introduction

The United States Department of Defense (DoD) is the largest user of modeling and simulation (M&S) applications in the world [Balci, 2001; Balci and Ormsby, 2002]. Though the first agent-like combat model appeared as a cellular automata simulation in [Woodcock, *et al*, 1988], agent-based combat simulation remains a relatively new and unexplored tool available to the DoD analytic community, but interest in this area has been increasing. This research extends agent-based simulation theory and knowledge and develops methodologies for DoD use of agent-based simulations. The intent is not to advocate wholesale adoption of agent-based simulations for the study of combat. Instead, the intent of this research is to conduct an initial, thorough investigation into their viability and develop methodologies and tools necessary for their proper application in combat analyses, particularly at higher levels of model aggregation.

An immediate question is what motivates undertaking this research? Human behavior significantly impacts the outcome of actual combat. However, removing the variability associated with the individual decisions within a heterogeneous group of

combatants has long been the practice of the military modeling and analytic community [Koopman, 1970]. The legacy models used by the DoD, therefore, fail to model and to capture the effects of diverse human behavior, known among the military analytical community as the intangibles [Bergeman, 2001]. As a result, there are many important aspects of combat that remain unexplored, their effects hidden from the military analyst and, ultimately, decision makers who use the modeling insights provided by the military analyst.

Outside the military analytical community, some of these same issues are being addressed through a relatively new modeling paradigm, agent-based simulation. A wide variety of fields including artificial life [Levy, 1992], artificial intelligence (AI) [Russell and Norvig, 1995], and social sciences [Holland, 1995; Axelrod and Cohen, 2000] have employed the tools of what has become agent-based simulations to investigate some of the dynamic effects of heterogeneous behavior.

As a tool for military decision makers, agent-based combat simulations similarly offer potential for exploring the impact of many aspects of human behavior on effectiveness in combat operations - insight beyond the scope of the established simulations due to the assumptions that homogenize combat participants and their behavior. Therefore, as a result of the successful application of agent-based simulations in other fields, interest in agent-based simulations is growing within the military M&S community. Champagne (2001c) details current issues in modeling human behavior specific to combat analysis with emphasis on agent-based modeling.

However, the majority of the research into agent-based systems is not directly applicable to modeling combat. The majority of the work in the field concentrates on cooperative agents [Sycara, 1998]. By their very nature, combat simulations are constructed to explore the effect of conflict. As a result, the academic literature exploring agent-based combat simulations is notably sparse.

Moreover, in spite of the potential for improved insight into the mechanisms of combat, the vast majority of the work in the area of combative agents has been in simulating small, toy problems and elementary scenarios that little reflect real-world combat. Project Albert, a U.S. Marine Corps (USMC) project dedicated to the advancement of agent-based simulations, refers to the state-of-the-art in agent-based combat simulations as “an intellectual sandbox” in which the most basic problems are explored through rudimentary scenarios [Widdowson, 2001].

In order to become a more relevant tool, agent-based simulations must demonstrate applicability on real-world scenarios beyond simple small force, engagement-level models. However, there remain a host of issues that must be studied before this can become a practical reality. A primary question is whether or not these agent-based methods are applicable to modeling mission-level scenarios. In making this determination, criteria must be developed to establish what “good enough” means for agent-based simulations. In fact, as the sheer volume of verification and validation literature attests, determination of what it means for a simulation to be “good enough” remains a serious issue for all combat simulations and is not unique to agent-based

simulations. Modeling in an agent-based paradigm does not in-and-of-itself cause this issue to disappear.

One ultimate goal of agent-based simulations is to provide capabilities to capture better the variability associated with human behaviors. An intrinsic problem in this goal is the lack of methodology for quantifying the characteristics governing human behavior. If agent-based simulations are to provide combat modelers with in-roads into the behavioral aspects of combat, agent-based modelers are faced with developing scientifically defensible decision-making algorithms to embed within the agents in these combat simulations.

1.2 DoD Simulations

As the world's largest user of modeling and simulation applications [Balci, 2001; Balci and Ormsby, 2002], the DoD has numerous types of simulations available, ranging from full live-fire exercises to virtual training environments to completely computerized simulations. Additionally, the DoD is becoming adept at integrating their simulation environments, thus providing aggregated simulations containing any or all of the above types of simulations. This research is focused on completely computerized simulations, commonly called constructive models.

1.2.1 Constructive Simulation Classification

The DoD generally classifies its constructive models into categories based on their level of data aggregation and their scope. Typically, there are five recognized model categories: engineering, engagement, mission, campaign (or theater), and macro-

levels (see Figure 1.1). The scope and level of data aggregation are highly correlated, and generally speaking, the broader the scope of the model, the greater the level of data aggregation. Furthermore, data from lower-level models are generally aggregated to provide data to higher-level models. Figure 1.1 and the accompanying discussion have been frequently presented in DoD simulation briefings including [ASC/XREWS, 1992; AFSAA, 2000; Champagne, 2000].

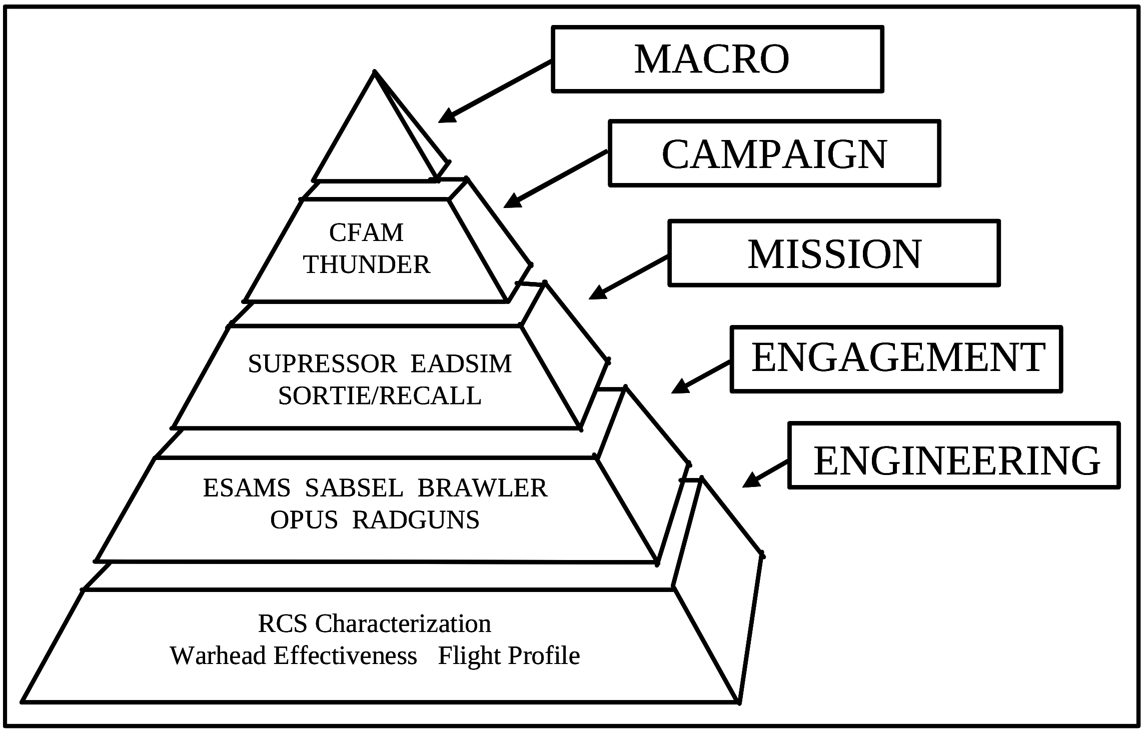

Figure 1.1 Modeling pyramid with representative models

Figure 1.1 depicts the DoD modeling pyramid for constructive models and the associated categories. As the categories move up the pyramid, the level of detail modeled decreases; the amount of data aggregation increases; and the scope of the models

increase. For readers familiar with DoD constructive models, specific examples are indicated at their respective levels within the pyramid for added context.

Each model within the engineering-level of the pyramid is specific to a single system or event. The data and model specifications are generally highly detailed and grounded primarily in scientific and physical laws and properties. An example of this type of model is a finite element model of an airframe or a detailed simulation of a missile's flight profile. The results of the model would be highly detailed as well and may include stresses on every element over time or flight parameters at each of many very small time increments.

The level above engineering contains the engagement-level models, typically described as "few-on-few." Data in engagement-level models are less detailed than in the engineering models, while the amount of aggregation is increased. An example of an engagement-level model might be a simulation of a sortie of four aircraft attacking a defended target. In such a simulation, the flight paths, radar cross-sections, and weapon trajectories would still be highly detailed, but the damage computations are generally not computed in detail. Instead, the results of engineering models are usually aggregated to provide probabilities of damage given particular simulation conditions using techniques such as look-up tables or probability curves.

Mission-level models occupy the third tier of DoD constructive simulations, and these are often called "many-on-many" models. A typical mission-level model may simulate the air-to-ground engagements on the first day of a simulated conflict. Again, the results of engagement-level models may be aggregated to provide inputs to these

mission-level models. For example, an engagement model outcome of some strike package attacking a defended target may be summarized as expected outcomes in the mission-level model.

Campaign-level models are highly aggregated. Such models may employ a playing field that comprises an entire country and may simulate days, months, or even years of combat. Furthermore, these models most often simulate joint or combined service activities in the region. Almost all such models use data aggregated from one or more of the models found lower on the pyramid as inputs, typically providing various effectiveness data.

Macro-level models occupy the top tier on the modeling pyramid, and these models typically contain the most aggregated conceptual models and supporting data. These tend to be special-purpose or spreadsheet-type models used to estimate force level trends. These are not as widely used (or accepted) as the models comprising the four lower levels due to the many overly-broad assumptions necessary to reduce campaign-level combat to a few number of spreadsheet calculations. Macro-level models tend to be very specialized models, functioning in many cases as a modern form of the "back-of-the-envelope" analysis. A typical use may include identification of potentially promising scenarios to study using a more extensive campaign-level simulation or determining a rough estimate of a desired weapons system fleet size.

The current state-of-the-art with respect to agent-based combat simulations resides in the area of the engagement-level models. The most advanced of these simulations involve small numbers of combatants and short time spans. However, unlike

the legacy models occupying this tier of the modeling pyramid, the agent-based combat models do not use detailed data and do not provide a methodology for validating their results against real-world scenarios. Linking results from agent-based combat simulations to the real-world remains an elusive target for military analysts.

1.2.2 Agent-Based Simulation

Software agents are autonomous entities (objects) within a virtual environment and are an outgrowth of the Object Oriented (OO) software design paradigm. Agent-based programming holds many of the promises of OO design, such as reusability and ease of maintenance. Additionally, agents have been shown to be particularly advantageous on open and distributed systems [Sycara, 1998]. Agent-based software has a strong emphasis in the recent literature and has been successfully employed in many different environments and for many differing purposes. Agent-based simulations are stochastic models with software agents comprising the bulk of the model.

Though the employment methods of agent-based simulation have their roots in OO design, the concepts grew from early work in the fields of artificial life and artificial intelligence (AI). These fields are primarily concerned with entity behavior and entity interaction rather than with the performance of a particular system. That is not to say system-level performance is not of interest; instead, the system-level performance is a phenomenon growing out of individual behaviors and interactions rather than the focus of model construction. This bottom-up focus is a real paradigm shift for most simulation modelers.

Traditional modeling methods use a top-down approach in building a system-level model. Assumptions are made about how the system works, most often taking the form of quantitative or logical relationships that then make up the conceptual model of how the system behaves [Law and Kelton, p. 5, 1991]. Agent-based simulation, on the other hand, is primarily concerned with the behavior of the entities that make up the system. Assumptions about the behavior of these entities within a particular system constitute agent-models, which then act within a specific environment. System level behavior emerges from the actions, or inactions, of the various agents within the environment. In this way, agent-based simulation is a bottom-up approach to system model development.

To distinguish between system-level specification and emergent system behaviors, consider the following. Law and Kelton (p. 106-107, 1991) describe a bank modeling process in which "one might collect interarrival times…" to specify interarrival-time distributions for the model. In such a model, the simulation would explicitly specify a distribution for entity arrivals. In Chapter IV, this research presents an agent-based model where the agent-arrival times are an emergent phenomenon derived from the agent behaviors. Though this emergent system behavior conforms to historical assumptions about the system, it is not a predetermined system specification.

The emergence of system-level behaviors from the interaction of individual entities is one of the defining characteristics of agent-based systems, a phenomenon known as emergent behavior [Holland, 1995; Russell and Norvig, 1995; Axelrod and Cohen, 2000; Bonabeau, 2002]. Specifically, emergent behavior is system-level behavior, not specifically programmed into the simulation, resulting from the behavior of

entities within the system. In complex system analysis, where system-level outcomes are highly dependent on entity interaction, agent-based simulations are being used to discover the mechanisms of individual behaviors that create or avoid specific emergent behavior [Levy, 1992; Holland, 1995; James, 1996; Axelrod and Cohen, 2000].

An attractive feature of emergent behavior is that it allows models to capture known behavior that generally defies analytical explanation. For example, as described in Bonabeau (2002), one agent-based model's emergent behavior demonstrated Braess' paradox, which describes the counterintuitive worsening of traffic congestion when an extra lane is added to a transportation network. An agent-based model can also augment theoretical results by extension beyond the limitations of the theory. For example, Champagne, *et al*, (2003) and Carl (2003) replicated theoretical search results, but then extended search theory to include overlapping search, which demonstrated that the overlapping search could produce better results than the more efficient, non-overlapping search.

Agent-based simulations have recently emerged as an area of interest in the arena of combat modeling. The autonomous nature of software agents gives them a natural niche in the distributed models used for wargaming and training. Additionally, the ability to encapsulate the behavior mechanisms for each agent within the object suggests that this paradigm offers a chance to study the effects of individual behaviors on combat effectiveness, aspects of combat not captured previously in constructive simulations used for combat analysis [Ilachinski, 2000].

This method of creating combat participants suggests that the effects of individual behaviors can be studied. Similarly, the literature suggests that the effect of the value systems of the combatants on the outcome of combat are now open for study by the military analyst. Therefore, agent-based combat simulations *promise* to allow unprecedented insight into factors governing the outcome of war that have been inaccessible previously, other than in doctrinal musings. This agent-based paradigm promise will only be realized once the nuances of combat agent-based modeling are investigated, understood, and appropriately applied.

1.3 Research Goal

This research was funded by the Defense Modeling and Simulation Office (DMSO) and the Air Force Research Laboratory/Human Effectiveness Directorate (AFRL/HES) to investigate the possibility of advancing the state-of-the-art in agent-based combat modeling on several fronts. In support of this goal, various objectives were established. Those objectives are discussed below.

1.3.1 Establishing the Background and Supporting Work

Military analysts are increasingly looking for inspirations from the fields of Chaos and Complexity as they search for additional tools to study factors governing combat effectiveness. Work in AI, artificial life, and complex adaptive systems (CAS) suggests that many effects influenced by human behavior can be successfully modeled using agent-based simulations. As a result, agent-based simulation may provide insight to crucial aspects of combat not currently modeled by the legacy models. Champagne

(2001a) traces work in these fields of Chaos, Complexity, and artificial life as they pertain to modeling combat as a CAS, and Champagne (2001c) details issues in organizational and human behavior relevant to combat modeling.

1.3.2 Extend Agent-Based Combat Simulations to the Mission-Level

Agent-based combat simulations to date generally suffer from a failure to connect the modeled scenarios to real-world combat scenarios. The vast majority of agent-based combat modeling has focused on rudimentary scenarios, relying on broad extrapolation of insights to more complex scenarios [Widdowson, 2001]. Though these efforts are providing some useful analytical insights into combat, broad acceptance of analytical insights will come only when these models prove to be capable of providing relevant insights into more substantial real-world situations. This research proposes to extend the agent-based modeling paradigm to model a WW II combat operation. The purpose of this is to extend the state-of-the-art in agent-based combat simulations to encompass the mission-level of the modeling pyramid (Figure 1.1).

With respect to this research objective, specific contributions of this work include: definition and demonstration of a mission-level agent-based modeling tool and a methodological approach to defining and building an agent-based model based on historical combat.

1.3.3 Develop Validation Methods for Agent-Based Combat Simulations

In extending agent-based combat simulations into the mission-level of modeling, techniques for determining the extent of model correctness are crucial in developing

useful applications. This research extends verification and validation (V&V) techniques to agent-based simulations. This includes developing a taxonomy for V&V techniques currently absent from the simulation literature as well as a quantitative methodology for assessing agent-based model validity.

With respect to this research objective, specific contributions of this work include: development of a taxonomy for both verification and validation treating each component as a separate, but related, function in a comprehensive process; and extending the verification and validation taxonomy to accommodate agent-based models.

1.3.4 Demonstration of Methods via Known Use-Case

In pulling together the results from the above research, it is important to demonstrate agent-based techniques through the development of a mission-level model reflecting a relevant real-world military scenario. Therefore, another objective of this research is to develop a scenario based on the Allied offensive search for U-Boats in the Bay of Biscay during World War II. The Bay of Biscay agent-based simulation is then used as the basis for experimentation in support of the theoretical work advanced through this research.

While there have been historical studies using agent-based simulations, primarily under Project Albert, little scientific rigor has been applied to: 1) determining and parameterizing the underlying behaviors; 2) researching the model parameterizations required for historical accuracy; and 3) quantifying the sufficiency of the model behavior with respect to the historical record. Such rigor must be established for agent-based

combat simulation to gain a respected foothold in the military modeling and simulation community.

With respect to this research objective, specific contributions of this work include: encapsulation of an historic combat scenario into an agent-based model; demonstration of extended verification and validation taxonomy; and demonstration of statistical methods useful for assessing model behaviors.

1.4 Contributions of this Research

This research makes several contributions, which are summarized below.

The state-of-the-art in agent-based combat simulation is established through a comprehensive review of the literature. This review delineates the strengths and potential weaknesses of agent-based models particularly as compared to legacy modeling approaches.

In extending agent-based simulation techniques to the mission-level, agent-based combat simulations are extended to address real-world military scenarios. In showing the veracity of the proposed simulation, additional contributions are made to simulation V&V. Primarily, a taxonomy of verification and validation techniques is developed, to include methods of validating agent-based simulations, and output analysis techniques were extended to incorporate the validation of emergent behavior in the agent-based model.

Finally, a novel statistical validation methodology was developed to determine model veracity with respect to the stochastic process underlying the real-world combat

operations. The technique combines two nonparametric techniques, the bootstrapping and sign test, to enhance the information available through the use of more traditional methods such as the t-test.

1.5 Sequence of Presentation

The remainder of this document is comprised of five chapters. Chapter II provides the necessary background on the agent-based modeling paradigm and reviews the relevant literature concerning agent-based combat simulation. Chapter III reviews the V&V literature and presents a new taxonomy of V&V techniques and a methodological approach for applying these techniques within a modeling and simulation process. This includes extensions to agent-based models. Chapter IV details the development, verification, and validation of the Bay of Biscay agent-based simulation. Additionally, a basis for extension of this historical scenario into modern national security scenarios is presented. Chapter V develops a new statistical approach to validation of combat simulations based on historical data. Finally, Chapter VI summarizes the contributions of this research and proposes areas for future efforts.

II. Agent-Based Simulation

Agent-based software is a natural extension of the object-oriented paradigm. Agents are generally objects that extend the concept of modularity to the point where the objects behave as autonomous entities. Therefore, agents are a subset of objects, and while agents are objects, not all objects are agents. Moreover, having their behavioral methodology internal to themselves, agents provide an innate metaphor for natural systems. In a combat scenario, it is easy to envision self-encapsulated software objects (agents) representing the combatants.

The power of agent-based software comes from the ability of agents to interact with other agents as they seek to fulfill their internal goals. When there are many interactions between agents, the system often exhibits emergent behaviors typical of Complex systems. Emergent behavior is system behavior not specifically programmed (intended or unintended). Moreover, being self-contained, the agents are extremely well suited for operating in open and distributed systems. Each of these properties receive more detailed attention in the subsequent sections of this chapter.

This chapter defines the terms “agent” and “agent-based simulation.” Relevant background to agent-based systems is provided through examples in the literature. Finally, the state-of-the-art with respect to agent-based combat simulation is presented, highlighting deficiencies within past agent-based approaches that are addressed in this research effort.

2.1 Agent Defined

Within agent-based programming, the term “agent” has undergone a blurring of definition and has become somewhat ambiguous in modern software terminology [Sycara, 1998]. As more research funding goes toward agent-based technologies, the natural tendency for researchers is to broaden and stretch the definition of an agent to increase their chances for funding [Hendler, 1999]. Therefore, it is important to clarify what is meant by “agent” before discussing how they fit into an agent-based simulation.

Agents have been written about in the literature since the late 1980s and represent hardware, software, or some combination, existing in and interacting with a real or artificial environment. In its most basic definition, an agent is defined as anything capable of perceiving its environment and acting upon that environment [Russell and Norvig, 1995]. Such a broad definition means a host of scientific/academic communities can use “agents” in their research, resulting in a confusion of terminologies and multiple research area threads that tend to blend together [Hendler, 1999]. Indeed, under this broad categorization, there can be little distinction between simulation entities common in discrete event simulations (DES) and more recent concepts of agents found in the literature.

In this research, the definition of agents is more restrictive and mirrors the consensus of the agent-based systems literature. This research concentrates on constructive simulations (i.e. completely computerized simulation environments), thus an agent is limited in this context to a software entity. An agent, therefore, is a software system, situated in some environment, capable of flexible autonomous action to meet its

design goals within that environment [Jennings, *et al*, 1998]. This definition contains three key characteristics: situated, autonomous, and flexible. These characteristics provide the distinction between agents and other software entities.

Situated requires the agent to receive sensory input about the environment. Moreover, the agent must be able to affect this environment through its actions. Since agents are capable of both sensing and affecting their environment, many other AI configurations, such as expert systems, are precluded from being classified as agents [Russell and Norvig, 1995; Jennings, *et al*, 1998].

Autonomy requires that the agent should be capable of acting without direct, outside intervention. More specifically, agents have their own independent thread of control [Jennings, *et al*, 1998], so the agent should have control over its own actions and internal states. Autonomy is the characteristic that provides differentiation between "objects" and "agents."

Flexibility is the final characteristic differentiating agents from other software constructs. Flexibility, in turn, is defined in terms of three attributes: responsiveness, pro-activity, social ability. *Responsiveness* is the ability to respond in a timely manner to perceived changes in the environment. *Pro-activity* is the degree to which the agent exhibits goal/utility directed behavior. Finally, *social ability* is the degree to which an agent is capable of interacting with other agents [Russell and Norvig, 1995].

There are other agent-defining characteristics proposed to varying degrees by other researchers. For instance, in open architecture and distributed systems such as the internet, mobility is often touted as an important agent characteristic. In other

applications such as agent-based route planners or heuristic search applications, adaptability is often stressed. However, though particular applications of the agent-based system may require additional characteristics to be most effective, the core agent characteristics – situated, autonomous, and flexible – remain to differentiate between agents and other constructs.

2.1.1 Differentiating Between Discrete-Event, Object-Oriented, and Agent-Based Simulations

Discrete-event, object-oriented, and agent-based simulations are at their core simulations. The distinctions between discrete-event, object-oriented, and agent-based simulations do not lie in their component functions. Instead, how the simulation components are treated (implemented) from a programming standpoint distinguishes these simulation types. The implementation specifics of the simulation components do not necessarily give one simulation type abilities or functionality that cannot be ultimately engineered into the others. However, the design implementation may allow easier (or harder) simulation of some environments or systems than would be the case under another simulation paradigm. As an analogy, consider that many different computer programming languages will allow a programmer to accomplish identical tasks. However, some languages, through their design focus, allow some tasks to be accomplished more easily through one particular language than through others. For instance, graphical user interfaces can be developed in FORTRAN but are much easier to create in Visual Basic, a language specifically built to facilitate graphical design. In

simulation applications, one could develop a simulation of a manufacturing plant using C, but will likely accomplish the task more easily using SIMAN, SLAM, or SIMSCRIPT.

In delineating between these simulation paradigms, it is helpful to determine their commonalities first. Law and Kelton (1991) list and define the following components of a discrete-event simulation model:

> System state: The collection of state variables necessary to describe the system at a particular time
>
> Simulation clock: A variable giving the current value of simulated time
>
> Event list: A list containing the next time when each type of event will occur
>
> Statistical counters: Variables used for storing statistical information about system performance
>
> Initialization routine: A subprogram to initialize the simulation model at time zero
>
> Timing routine: A subprogram that determines the next event from the event list and then advances the simulation clock to the time when that event is to occur
>
> Event routine: A subprogram that updates the system state when a particular type of event occurs (there is one event routine for each type of event)
>
> Library routines: A set of subprograms used to generate random observations from probability distributions that were determined as part of the simulation model
>
> Report generator: A subprogram that computes estimates (from the statistical counters) of the desired measures of performance and produces a report when the simulation ends
>
> Main program: A subprogram that invokes the timing routine to determine the next event and then transfers control to the corresponding event routine to update the system state appropriately. It may

also check for termination criteria and invoke the report generator when the simulation is over.

Banks, *et al*, (1996) gives a similar list of components with several additional delineations, including:

Entity: An object or component in the system which requires explicit representation in the model.

Attributes: The properties of a given entity (e.g. the priority of a waiting customer, the routing of a job through a job shop.

Additionally, Banks, *et al*, (1996) adds event scheduling to the function of the timing routine. Regardless of implementation, these components constitute and define discrete-event simulations. Object-oriented and agent-based simulations possess the same component functions but require particular implementation paradigms. Additionally, agents are objects, but with additional constructs that further distinguish them from the broader classification of objects. The important distinctions are characterized below.

<u>Entity representation</u>. In every discrete-event simulation, entities are characterized by a collection of attributes that completely describe the state of the person or thing as it is represented in the model at a given time. Under the object-oriented and agent-based paradigms, these attributes are grouped together and encapsulated within a single software module, called an object or agent, respectively.

<u>Data and data access</u>. Discrete-event simulations typically make use of common memory (to include named memory as found in FORTRAN). Access to the values stored in the memory is available to all procedures or functions sharing the same scope (i.e. global, procedure or function specific, etc.). In object oriented models, the data is

encapsulated within the objects and accessible only via defined interfaces (methods) within the object [Deitel and Deitel, 2002]. Within an agent-based paradigm, the data is also encapsulated, but the agent does not have to honor a request for data access [Sycara, 1998].

Event scheduling and entity actions. In both discrete-event and object-oriented simulation, there is a master schedule, called the *event list*, that sequences when events will occur. The event routines are typically subroutines or separate modules that are called based on logical processing of event list flags. In object-oriented simulations, the event routines associated with a particular entity type are contained within the object's methods. Each object must schedule its next event for some time in the future (or have some other related event schedule it) for that event to occur.

Within an agent-based simulation, the agent is running on its own thread of execution. As a result, there is no master sequencing. Instead, the agents request permission to act from the main simulation program based on the simulation clock time. Each requesting agent that needs to act at a discrete point in time is provided a slice of CPU time in which to perform their actions. As a result, the main simulation program does not necessarily know the event types that may occur within the simulation, only that an event will occur.

As an example of the agent-based approach, consider the Bay of Biscay agent-based simulation presented in Chapter IV. The simulation clock is kept and updated in the environment object, which serves as the main program. Once the simulation instantiates (creates) the agents and starts their individual threads at simulation start, the

agents internally schedule their next action. The main program has no indication of when a particular agent has an event scheduled. Instead, each agent notifies the simulation timer of its next event time and requests permission to act accordingly. If the agent's next event is scheduled for the current simulation time, it is given permission to act (i.e. let the event happen). If the next event is at some time future to the simulation clock, the agent is told to wait (i.e. not take action on the event). The timing routine notes the smallest future event time as the agents request permission to act, and the simulation clock is advanced to the next known event time. All agents are then notified that the simulation clock has been advanced so they may again request permission to act.

The object-oriented and agent-based approaches to building a particular simulation model have both advantages and disadvantages. Object-oriented design (and agents are objects) provides a "natural and intuitive way to view the design process – namely, by modeling real-world objects" [Deitel and Deitel, 2002], providing a natural way to conceptualize many real-world systems. As a result of following an object-oriented or agent-based approach, maintainability is enhanced through their naturally modular structure (a good software engineering practice). Additionally, when [Law and Kelton, pp. 103-105, 1991] distributed simulation is discussed, object-oriented and agent-based programming represent a logical, intuitive method for submodel decomposition for distribution over several processors. Indeed, agent-based systems in particular are well adapted for open network computer environments such as the internet or World Wide Web [Sycara, 1998]. As the DoD, in particular moves toward the High Level Architecture (HLA) for federated (open network) wargame simulations [*Modeling Human…*, 1998], agent-based applications present an attractive implementation avenue.

On the other hand, for non-distributed applications especially, agent-based simulations require more overhead to control the proper timing of agent-driven events. Additionally, the agent-driven events cause the main simulation program to give up control and authority over the simulation events. This is of particular concern in open network computing environments where the simulation designer may not have control over simulation agents implemented by other parties.

2.1.2 Differentiating Further Between "Agents" and "Objects"

Agent-based programming is an outgrowth of object-oriented (OO) programming, so agents and objects share some important characteristics. An object is a self-contained software entity (i.e. internally maintains all of its state data and methods for performing actions or computations). Important distinctions between agents and objects include autonomy and flexibility.

In object-oriented programming languages, objects can be programmed with varying levels of autonomy through the use of access modifiers (e.g. in JAVA® these are *public*, *protected,* or *private*), which can restrict access to their variables or methods. Variables and methods declared as private may only be accessed from within the object itself; protected limits access to other objects within the same package; and public allows unrestricted access. By maintaining private methods and variables, an object maintains control over its internal state. Such an object exhibits autonomy over its state [Jennings, *et al*, 1998].

An object cannot exhibit control (autonomy) over its behavior. Objects do not have their own thread of control, and an object cannot be (entirely) constructed of private

methods and still be useful. Some methods must be made available to other objects, or an object-oriented system does not function. Once a method is made publicly available, then it can be invoked at any time from outside the particular object. Therefore, the object has no control over when the method is invoked. Agents, on the other hand, function on their own thread and, as a result, maintain control over their state and behavior.

Flexibility also differentiates between objects and agents. The standard object-oriented model does not prescribe building responsiveness, pro-activity, or social ability into the system [Jennings, *et al*, 1998]. Though objects can be built such that these characteristics are integrated into the design to one degree or another, the standard OO program does not imply the presence of any of these characteristics.

2.1.3 Types of Agent Behavior

The primary contributor to the study of agents has been the field of AI. The study of intelligence, especially AI, is broadly categorized into four fields of study dealing with combinations of methods of thinking and acting (Table 2.1), and typically software agents used in the study of social sciences encompass one of these four areas.

Table 2.1 Four Categories of AI Study

	Human-centric	Rationality
Thought process	Systems that think like humans	Systems that think rationally
Behavior	Systems that behave act like humans	Systems that act rationally

[Russell and Norvig, 1995]

For systems built to think like humans, the focus is on cognitive modeling, or simulating the process of thinking as it is done in the human mind. Systems that are built to behave like humans concern simulating machines capable of passing a Turing test. Systems that simulate rational thinking are concerned with the logical process of arriving at a correct conclusion given correct premises. Finally, systems built to act rationally are geared toward producing actions that best achieve a set of goals given a set of beliefs. Most agent-based simulations fall within this latter category. Agents built under this construct are called rational agents.

This research is limited to the field of rational agents. Rational agents are intelligent agents that "do the right thing" [Russell and Norvig, 1995; Hendler, 1999]. Rational agents perform those actions producing the most "success" based on its goals and present knowledge (i.e. rational agents look for oncoming traffic before crossing a street because not getting run over improves its chances of getting to the other side). This characteristic makes them ideal for conveniently explaining many behaviors [Hendler, 1999].

A future avenue of research for agent-based combat modeling is the modeling of "irrational" combat agents (e.g. suicide bombers). Such models might then expand the space of potential combat outcomes from the model thereby improving overall levels of analytical insight.

2.1.4 Agent-Based Programming Defined

An agent-based program is one in which the primary abstraction within the system is an agent. For example, in a combat-oriented agent-based simulation, the role of

the agent is that of an individual component of the system such as a soldier, tank, aircraft, or ship. Each agent within the system is an autonomous entity that contains its own decision and action algorithms for use in its environment.

Agent-based systems can represent how natural systems work by distributing a problem among a number of autonomous entities [Middelkoop and Deshmukh, 1998]. Agent architecture is particularly useful when a problem can be readily decomposed into multiple sub-problems [McDonald and Talbert, 2000]. This is especially true when there is a great deal of parallelism possible; each agent is simultaneously performing its individual task [Moscato, 1999]. Additionally, the learning and adaptive nature of agents lends itself readily to problems containing uncertain situations [Middelkoop and Deshmukh, 1998], especially those systems that are prone to localized failures of some sort. Examples of such systems include natural processes (predator-prey), game theory, social sciences, political alliances, warfare, and other chaotic systems to name a few.

However, agents are not ideal for all problem situations. In particular, agent-based programming is not well suited for situations where a problem cannot be effectively divided into a series of interacting sub-problems or sub-goals. Similarly, if the desired actions are known and fixed, then the agent-based approach is not generally justified. In these cases, the high overhead associated with agent-based approaches is not warranted [Middelkoop and Deshmukh, 1998].

Because the agents are autonomous entities possessing their own decision and action algorithms, the purpose of the simulation mechanisms then is to establish the simulation environment, to start, to monitor, and to end the simulation, while collecting

pertinent data throughout. By analogy, suppose the agent-based combat simulation were considered a game. In a game, the agents would be considered the players, and the environment would be considered the field, court, or playing board. The simulation mechanisms would be the arbitrator (referee), who controls the start and end of the game and determines the winner based on the games rules.

2.1.5 Properties of Agent-Based Systems

As computing systems and applications become more complex, there is an increasing need for tools to handle the complexity. Two powerful tools for effectively handling complexity are modularity and abstraction [Sycara, 1998]. Agent-based systems offer both, when properly constructed. As a result, agent-based systems offer many potential benefits.

The primary property of agent-based systems is emergent behavior. Emergent behavior is not behavior that is explicitly programmed into the system. Instead, it arises as a consequence, sometimes unforeseen, of the myriad interactions between system agents. In many cases, emergent behavior is a benefit, enabling agents to collectively solve problems that they individually could not solve.

As a direct result of the emergent behavior phenomenon, agent-based systems have the ability to solve problems that are larger than the agents can solve on their own. The result is a loosely coupled system of problem solvers that locally solve a portion of the problem and then interact to resolve the tasks into the required solution. This brings some ancillary advantages as well. Primarily, by enabling a decentralized approach to problem solving, this alleviates the need for a centralized agent that monopolizes the

resources of a given location. This in turn reduces the risk of resource bottlenecks and protects against a centralized system that could fail at a critical time.

However, emergent behavior can take a form that is counter productive or even fatal to the system, meaning agent system designers must take special care in avoiding these types of emergent behavior, or at least building in specific mechanisms to identify and control the behaviors. Therefore, while emergent behavior is a powerful aspect of agent-based systems, it can also bring about unexpected and unwanted consequences. This is of particular concern in research avenues investigating autonomous swarms of unmanned aerial vehicle agents [Guadiano, *et al*, 2003].

In addition to (beneficial) emergent behavior, there are advantages to designing an agent-based system that are naturally derived from agents' roots in object-orientation and from their modular nature. First, modularity aids in the ability to decompose system development into small, easily managed tasks that can be handled by simple agents. Additionally, modularity also assists in easing the maintenance effort of the system components. Changes to an agent are made directly to its encapsulated data and methods (versus data and modules scattered throughout the simulation).

Aided by the flexibility of agents to dynamically reorganize in the system to solve new problems, agent systems can require less redesign. This holds for two reasons. First, it is a natural advantage stemming from the object-oriented nature, especially with respect to the inheritance property, of agent design. Second, once deployed with the proper interface, the same agent can be used by multiple applications to solve different problems for which their area of expertise is a necessary part.

Agent-based systems also have the ability to save a great deal of money for owners of existing legacy systems, those developed long ago and having critical functionality. Redesign of these legacy systems for use in an increasingly distributed environment is often extremely costly, if not impractical. However, system designers have the ability to "wrap" an agent around these legacy codes enabling the legacy system to remain viable in a distributed environment [Woods and Barbacci, 1999]. Wrapping entails constructing agents that function as front-end modules to the legacy code, or as intermediaries between two incompatible legacy systems. The agent then performs the necessary translations of data, input, and output to provide continuing serviceability to legacy systems without expensive redesign.

Agent-based systems also offer the chance for enhanced system performance in a number of ways. First, agent systems offer an opportunity for computational efficiency because simple, focused agents can work concurrently on their area of expertise without competing for centralized resources. This is true provided communications are kept to a minimal level. Second, agent systems provide added system reliability by introducing redundant capabilities. Agents can dynamically find alternate agents to accomplish specific tasks when other agents fail or are not present (in the case of open systems). Third, agent systems are capable of exhibiting an extensibility of resources in solving certain problems. This occurs when a number of agents and their various capabilities can be enlisted to work the same problem. Finally, agent-based systems are capable of a robustness not typically found in other systems. Through their very design, agents are capable of working in uncertainty and in a dynamic environment (e.g. search agents

situated on the world-wide web). This means that agents can handle anomalies locally without propagating them through the system.

There is an additional benefit agent-based systems provide as a direct offshoot from their design. Because the agents' sensors and behavior mechanisms are completely encapsulated within the object structure, they provide a natural metaphor for the real-world system. That is, it is easy to view the real world agent in terms of the virtual agent. Because of this, agent-based systems are particularly apt for providing solutions to problems that are naturally regarded as a society of autonomous interacting components.

The benefits that agent-based systems promise come with many challenges as well. Though research into multi-agent systems is advancing rapidly, the majority of agent systems are single agent systems [Sycara, 1998], and there are still many issues that must be addressed to fully capitalize on agent-based systems without falling prey to the disadvantages that such a loosely bound collection of software present.

One of the major concerns regarding agent-based systems is the lack of a centralized coordinating authority [Russell and Norvig, 1995]. Absence of a centralized controlling authority can allow unwanted emergent behavior, as previously discussed. The system developer must take great care to ensure that an agent system exhibits coherent collective behavior while avoiding unpredictable (or harmful) behavior. Moreover, the developer must be mindful that as well as avoiding harmful behavior, the nonlinearities associated with the agent interactions provide an environment that may be unstable, and the designer should take steps to avoid this consequence. Currently, a

centralized authority, of some form, provides the only sure way of handling harmful emergent behavior.

The decentralized nature of agent-based systems presents a lack of global control, perspective, and data. In this environment, the designer does not have the means to know what the state of the agents' coordination process is, and therefore, there is often no method for the designer to recognize and reconcile disparate intentions among the collection of agents attempting to coordinate [Russell and Norvig, 1995; Sycara, 1998].

Another challenge is the criticality of agent communications in multi-agent systems. Since agents work autonomously, a great deal of effort involves ensuring agents are able to request data and provide solutions correctly. Ensuring smooth communications between agents can be a major design undertaking, but it is essential to make sure agents interact correctly.

A big issue associated with inter-agent communication is the issue of resolving conflict and avoiding deadlock. Conflict occurs when two competing agents vie for the same resource. If conflict should occur, then the agents first must be able to recognize the conflict and then have methods for resolving that conflict. Deadlock, on the other hand, occurs when two agents are waiting for a response from the other before they perform some action. Under such a circumstance, neither agent will begin its required action. Special care must be given to removing all sources of deadlock in a multi-agent system when designing the agents. To further complicate matters, especially in open systems (e.g. the internet), consideration must be given to the interaction between heterogeneous agents that may be introduced into the system.

Perhaps the greatest challenge to agent-based systems is to design the system so that the agents are able to correctly formulate, describe, decompose and allocate the problems and sub-problems in such a way as to ensure that the agents are able to synthesize results from the system. A stable system with no unresolved agent conflict is of little use if the agents are not able to provide solutions for the problems they were built to address.

2.2 Types of Agent Systems and Uses

Agent-based systems have been used successfully across a number of different fields in recent years.

Of particular note is the success agent-based systems have had in heuristic optimization methods. Champagne (2001b) summarizes some recent agent-based heuristics based on population-centric models of natural systems such as ant colonies, immune system function, and swarming.

In addition to heuristics, the uses of agent-based software in the fields of networking and distributed computing are extensive, and well documented in the literature. As reliance on distributed systems increases, agents are being developed to monitor system performance, track component availability, and provide data on communication link performance with respect to the network [Sycara, 1998]. Usage of the internet is becoming dominated by agent applications in the form of "softbots," temporary agents performing specifically tailored tasks for each user of a site to customize searches and organize data [Hendler, 1999]. The number of agents seems to rival the number of potential tasks.

Agent-based software is finding particularly successful application with respect to data and information management. With disparate databases located throughout many distributed systems, agents are being assigned to place and retrieve data according to user needs. These agents can be used as “wrappers” that serve as an interface between otherwise incompatible systems, thereby alleviating the need for costly database conversions [Sycara, 1998].

McDonald and Talbert (2000) extended this concept for military simulation data management. They proposed maintaining a central repository of simulation input data using agent interfaces. These agents could be responsible for retrieving data and providing it to the user with the proper level of aggregation and in the proper format for the intended application. The net result would be the ability to maintain a single approved source of data for all military simulation uses, ensuring consistency between analyses and models. Though this is an extensive field of agent research with interesting application to the military analysis community, it is not a focus of this research effort.

2.3 Agent-based Combat Simulation

The first agent-based combat simulation to be found in the literature was a cellular automata (CA) model used to show tactics as an emergent behavior [Woodcock, *et al*, 1988]. Since then, as in many other fields of study, there has been increasing interest in the use of agent-based models for military analysis. In spite of a large and growing field of agent literature, most articles deal with cooperative agents, that is, agents with compatible goals [Sycara, 1988; Hendler, 1999]. In this aspect, work in the area of combat simulations differs from the vast majority of agent literature.

Perhaps the most coordinated effort to date at agent-based combat simulations is the US Marine Corps' Project Albert. This effort began with the idea of exploring "the middle ground between … highly realistic models that provide little insight into basic processes and … ultra-minimalist models that strip away all but the simplest dynamical variables and leave out the most interesting real behavior" [Ilachinski, 2000].

The first Project Albert simulation, Irreducible Semi-Autonomous Adaptive Combat (ISAAC), was built as a proof-of-concept model to demonstrate the applicability of complex adaptive systems (CAS) to combat modeling. Although ISAAC is often referred to as a "conceptual playground" [Ilachinski, 1998, 2000], it and follow-on simulations such as Socrates, Pythagoras, and Map Aware Non-uniform Automata (MANA) [Lauren, 2001, 2002] have demonstrated promise for gaining insights into battle not possible with traditional combat models. Published results have demonstrated the potential in ISAAC-type models to contribute in diverse areas such as the development of tactics as an emergent behavior [Ilachinski, 2000], exploring the role of combatants' trust in combat effectiveness [Bergeman, 2001], providing risk assessment for peacekeepers, and quantifying the value of reconnaissance to combat effectiveness [Lauren, 2001].

The models of Project Albert present a dilemma to the agent-based combat simulation researcher. Although the models employ many of the techniques of agent-based systems, the simulations are not strictly agent-based. For instance, the "agents" within these simulations do not have their own thread of execution. Therefore, the

entities within the simulations lack the requisite autonomy defining an agent. Instead, the simulations are categorized more accurately as object-oriented simulations.

In recent years, there have been an increased number of agent-based simulations used for studying various aspects of combat. For example, Tighe (1999), developed an agent-based simulation based ultimately on the boids flocking algorithm [Levy, 1992] and ISAAC [Ilachinski, 1998, 2000] as an attempt to find a method of quantifying strategic effects, purported to be one of the main strengths of air power in combat. Bullock (2000) continued the research into modeling strategic effects with the introduction of the Hierarchical Interactive Theater Model (HITM). This model was intended to provide a sufficiently complex tool able to show strategic effects of air power, while retaining enough simplicity to allow identification of interactions between important factors [Hill, *et al*, 2003b]. Other agent-based combat simulation research includes modeling riot tactics for small military units [Woodaman, 2000], small unit peacekeeping tactics in an urban environment [Brown, 2000], and a German training scenario involving small units over a relatively short time period [Erlenbruch, 2002].

Though each of the above are representative of the state-of-the-art with respect to agent-based combat simulation, Chapter IV outlines the development of an agent-based combat simulation based on the allied offensive against the German U-Boats in the Bay of Biscay during WW II and compares the model results with the historical data. This extends the state-of-the-art by validating the agent-based paradigm in modeling a significant real-world combat operation. This demonstrates that it is possible for agent-

based modeling to move beyond the "intellectual sandbox" and into more significant combat analyses.

2.4 Adaptation

Adaptive behavior is more sophisticated than emergent behavior in that experience provides the basis from which to select from alternative options and successfully meet new and diverse experiences. Adaptive behavior, therefore, is behavior that is formed as a result of the agents' experiences, and it provides a very powerful problem solving tool [Holland, 1995].

There are essentially two established avenues available for providing mechanisms that allow agents to change their strategies in adaptive systems, evolutionary or learning (and of course a combination of the two). Evolutionary strategies focus on exploiting the characteristics/actions that make up successful agents in a population and simultaneously providing a method for introducing new characteristics that may lead to more successful agent behavior [Holland, 1995]. Learning, on the other hand, derives future actions from prior knowledge gained from experience. Learning can occur through trial and error techniques or imitation of apparently successful agents. Additionally, learning may take the form of some type of supervised training [Looney, 1999].

Since the earliest CAS models were studied, genetic algorithm-type experiments showed that the interaction between populations of artificial species could produce individuals within the population that were especially hardy with respect to their environment [Ferber, 1999]. Although the internal structure of the individuals changed as a function of interspecies and environmental interactions, these individuals did not

display any real learning. That is, while the derived individual might be more “fit” for harsher environments, the changes could make the individuals unsuitable for the environment in which they were initially spawned, and this is especially true in co-evolutionary environments where competing agent-types are allowed to simultaneously adapt. Therefore, genetic algorithms do not provide an explicit mechanism for the retention of experience within either the population or the individual.

2.4.1 Types of agent adaptation

Though emergent behavior, in and of itself, is a potent characteristic of agent-based systems, the ability for individual agents to adapt to their environment gives these types of systems an additional (and powerful) tool that can be used to explore the system and its individual components. There are essentially two mechanisms of agent adaptation, learning and evolution.

2.4.1.1 Learning

No combat CAS simulation currently uses learning as a method of adaptation. Learning, however, is used in other agent-based applications and research. Learning can be done from scratch (i.e. no inbred knowledge), or it can begin from some predetermined, pre-programmed knowledge base. Enabling the combat agents to learn require mechanisms for allowing the agents to evaluate the success of their chosen strategies in any context in which they find themselves. This is generally done via an attribution of credit mechanism. Under such a mechanism, success due to any given course of action would receive a positive credit, thereby increasing the chance that the

same course of action will be performed again. Conversely, failure would result in a negative credit that discourages future selection of the same course of action.

While computers are astoundingly good at following algorithms, much more effectively than humans, in order to solve complex problems, they are far inferior to humans in the realm of learning. While humans are incredibly adept at applying their experiences to new situations, it is very difficult to get computers to adapt to even moderately different situations. Getting computers to learn and adapt has been a focus of AI and other branches of computer science almost since the advent of the computer [Russell and Norvig, 1995; Levy, 1992].

Examination of the mechanisms of learning gives great insight into the reasons humans exceed computers in their ability to adapt to new situations. Humans, it seems, learn by methods of abstraction, pattern recognition, and aggregation. These involve recognizing similarities between objects or events and classifying them based upon these similarities. Then when faced with a new object or situation, if these similarities are found, the same classification is applied. Once a sufficient knowledge base is built, subtleties can be recognized and sub-classification can be formed.

Anyone who has had a child can recognize the process. As an example, consider a child just beginning to speak and learn the names of objects. The ability to apply abstraction is well developed very early. For instance, a child might learn the word “bird,” and in the beginning, the word might be applied to most anything in the sky. Soon, however, the child will begin to recognize birds by their form and then accurately name birds that were stationary in trees. What’s more, the child would be able to identify

birds in pictures, too. In a short time, and with the correct experiences, the child might soon be capable of identifying owls as a particular bird, both in life, pictures, and drawings. Experience has demonstrated that in the child's knowledge base, there is an abstracted model of both generic bird and specific owl that allow him to correctly distinguish these in a wide variety of circumstances.

As simple as this seems, such a task for a computer is quite formidable. Recognizing birds and owls within a group of birds whether in real life, in pictures, and/or even simple drawings would require the programming of what humans would consider the "essence" of birds (and the almost infinite array of subtleties that further delineate owls from birds) into some sort of knowledge base. Then, when presented with an object, the computer would require the ability to abstract the object sufficiently for it to resemble the appropriate representation in its knowledge base. When the abstraction resembles two or more entries in the knowledge base, some decision process must allow the computer to select the most appropriate entry.

When applying experience to new situations, the process is very similar. Faced with a new situation, a human generally looks for ways in which the new situation matches any experienced previously. Indeed, the new situation may remind him of several different experiences simultaneously. To find the best course of action, the human would compare current goals to those it faced in the previous experiences and choose the path that experience has proven to be most effective given the likeness of goals. The process involves abstraction to a sufficient level to either draw from experience, or recognize that there is no previous experience from which to draw.

On the other hand, while abstraction is somewhat inherent in humans, computers have no inherent capabilities that are not explicitly programmed. More importantly, the subtleties of abstraction that make humans good at adapting to new situations are often functions of personality, including "gut feelings" or attitudes toward risk. These are not easily quantifiable in terms of software encoding.

2.4.1.2 Artificial Neural Networks

Artificial neural networks (ANN) are artificial intelligence approaches to learning used in discrimination and function approximation. Their name is derived from their (theoretical) structural similarity to neurons in the brain. A typical network consists of one or more hidden layers of "neurons," weighted functions, which respond to the input values according to an activation function that differs according to the type of network used and an output layer of neurons. The response is an adjustment of the weighting of the function.

In order to produce the discrimination or estimation function, the networks are given input data used to train the network. The training methods used differ according to the type of neural network employed, but the net result of the training is a series of weights that can then be used to approximate the underlying function of interest, assumed to have produced the data. When the weights have been adjusted to best fit the training data, the network has "learned" the process that produced the data.

2.4.1.3 Genetic Algorithms

Genetic algorithms are a common method of adapting software agents, and they are well established in the literature. GAs are so named due to their similarity to the

biological process of sexual reproduction in species used to generate the genetic makeup of the next generation [Holland, 1995; Axelrod and Cohen, 2000]. The generic process works in three steps: reproduction, recombination, and mutation.

The three steps of a GA together allow for an efficient heuristic search of the parameter space. Reproduction and crossover provide a method of intensification, searching heavily in areas shown to be good. Giving unsuccessful agents a (small) chance to influence the next generation of agents ensures that some parameters that would be good in combination with others are not entirely lost to the population. Mutation, on the other hand, provides diversification, the ability to search new areas. Together, these three simple steps concentrate on promising areas of the parameter space (intensification), while simultaneously allowing the search to escape local optima (diversification).

2.4.2 A New Approach to Agent Adaptation

Adaptation in agents occurs through any process that modifies agents' behaviors based on their experiences. Though the two approaches to adaptation, GAs and ANN are the most typical methods for agent adaptation, they are not the only ones. The complexity of computation associated with each of these methods and the volume of data required may be more than the modeler is willing to concede (or, in the case of data requirements, require more data than exists) during simulation execution. This research developed a different approach allowing a sufficient amount of adaptation to occur without incurring the computational or data intensity associated with GAs or ANN. The

adaptation algorithm is developed in Chapter IV. Results from the proposed method of agent adaptation are presented in [Price, 2003; Hill, *et al*, 2003a].

2.5 Conclusion

Agent-based systems are finding increasing acceptance in a wide variety of fields. However, until recently, the majority of the research has dealt with cooperative agents used in optimization heuristics, database management, and distributed network management. Agent-based simulation has only made in-roads into the modeling of combat in the last five years.

Agent-based systems are built on the premise that system-level behavior emerges from the interactions between the entities within the system. Rather than construct models that concentrate on the system, these models focus instead on modeling the individual system components and their behavior within the system. Under this paradigm, it is little wonder that social sciences, combat analysis among them, have become interested in utilizing these models to gain insight into effects of individuals' actions and decisions on real-world systems.

Much of the work touted as agent-based within the military M&S community does not approach the autonomy of system entities required by academic consensus to be considered truly agent-based. As for those combat models that are actually agent-based, most of the combat modeling to date has concentrated on exploring small, toy problems with little linkage to real-world scenarios that would establish legitimacy within the analytical community.

III. Simulation Validation and Verification Methodology and Taxonomy

According to the DoD, a model is "a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process" [DoDI 5000.61, 2002; DoDI 5000.61, 1996]. Balci (1994) defines a model as "a representation and abstraction of anything such as a system, concept, problem, or phenomena." Though V&V literature provides other various definitions, a common aspect runs through them all – that a model is a simplifying abstraction of some real-world system. The model then allows for experimentation or analysis by proxy when it would be impractical or infeasible for experimentation or analysis using the real-world system.

As an abstraction from reality, any model is, therefore, an imperfect representation of the real-world system it represents. In spite of imperfections, however, the use of models is an integral part of the decision making process, whether the model resides solely in the mind of the decision maker or is a more substantive, formal model constructed to specifically explore the implications of specific decisions or phenomena [Jenkins, Deshpande, and Davison, 1998]. The purpose of V&V is to provide tools and methods for determining the extent to which the imperfect model accurately represents the real-world system.

Though the concepts and terminologies have matured since the subject was first addressed almost four decades ago, many of the underlying problems associated with V&V remain. Naylor and Finger (1967) write "management scientists have had very

little to say about how one goes about ‘verifying’ a simulation model or the data generated by such a model” and “the reason for avoiding the subject of verification stems from the fact that the problem of verifying or validating computer models remains today perhaps the most elusive of all the unresolved methodological problems associated with computer simulation techniques.” Other authors have noted these weaknesses as well. For example, Schrank and Holt (1967) wrote, “the validation problem has been neglected” and “even though the methodology of validation is still so undeveloped, it is critically important that serious and extensive efforts be made to test and validate simulation models before applying them.” Naylor and Finger (1967) further address the significance of V&V when they write “verifiability is a necessary constituent of the theory of meaning. A sentence the truth of which cannot be determined from possible observations is meaningless.” More recent literature underscores the same general weaknesses in the field. Kleijnen (1996) points out the lack of a standardized general V&V methodology when he writes “unfortunately, the literature gives neither a standard theory on validation, nor a standard ‘box of tools’.”

The purpose of this chapter is to address this lack of standard theory in the validation literature. This chapter consolidates current definitions, develops a taxonomy of V&V techniques, and extends V&V into agent-based models for the first time. The V&V methodology is outlined based on several current models of the overall modeling and simulation process.

3.1 Definitions

Early V&V literature did not distinguish between verification and validation functions; instead, all techniques used to determine a model's correctness, applicability to an application, and scope of applicability were commonly grouped together under the term *model verification* [Naylor and Finger, 1967]. However, verification and validation functions were soon made distinct. Mihram (1972) proposed a five step modeling process (adapted for Figure 3.2), which included verification and validation as separate steps. More recent literature [Law and Kelton, 1991; Balci, 1994; Banks, Carson, and Nelson, 1996; Kleijnen, 1995a; Kleijnen, 1995b; Kleijnen, 1996] maintains the distinction between the two modeling functions (i.e. V&V) in determining overall model fitness.

There are many verification and validation techniques available for building confidence in the results produced by a model, but there is no standard set of tools applicable to all models. However, no technique, or set of techniques, can prove beyond all doubt that a model is entirely correct [Forrester and Senge, 1980; Balci, 1994]. Instead, each successful test is intended to provide an added measure of surety with respect to the accurateness of the results produced by the model [Naylor and Finger, 1967]. Similarly, a failed test does not completely "invalidate" a model. The failure merely highlights a shortfall in the model's range of applicability [Hodges and Dewar, 1991]. The extent to which this failure impacts the model's usefulness is, in the end, a matter for the model user and is influenced by the risk imposed in using a model that potentially produces harmful results.

Any suitable V&V taxonomy requires unambiguous terminology. To this end, the remainder of this section defines the important concepts in both the M&S and V&V processes.

A *conceptual model* is the abstraction of the real world system [Balci, 1994]. The extent to which it is an accurate representation is determined by the techniques used to verify and validate the implemented model. Though the majority of the literature deals specifically with computerized simulation models, most of the definitions and techniques are applicable to implementations extending beyond the computer simulations. Indeed, since computer programs are algorithmic, the principles must necessarily apply to any implementation of these algorithms, regardless of the implementation environment.

Law and Kelton (1991) define the process of model *verification* as "determining that a simulation computer program performs as intended," and many publications in this field subscribe to this definition [see Kleijnen, 1995a, 1995b; Page, *et al*, 1997]. Verification ensures that the executable model is built correctly. Verification does not indicate the correctness of the conceptual model or the aptness of its implementation; instead, it is the process of determining the accuracy of the implementation of the conceptual model within the chosen modeling environment. This process is generally referred to as debugging and is primarily concerned with finding and correcting syntactical and logical errors in model implementation. Verification, therefore, ensures that the conceptual model is correctly and faithfully implemented in the executable model.

Law and Kelton (1991) define *validation* as “concerned with determining whether the conceptual simulation model … is an accurate representation of the system under study.” Others cite Schlesinger, *et al* (1979), who defines validation as “substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model” [see Sargent, 1991, 1996; Balci, 1994, 1995; Balci and Sargent, 1984; Fraedrich and Goldberg 2000]. Validation, therefore, is concerned with building the right model for its intended application. Likewise, validation techniques are used to provide confidence that the conceptual model sufficiently represents the real system being studied and that the implementation of the conceptual model is sufficient for the purposes of the particular study being conducted.

These definitions (verification and validation) taken together indicate the building of user trust to a necessary level of sufficiency relative to a specific application. Therefore, a model should be developed for a specific purpose or application, and its applicability, likewise, should be determined within the context of that purpose [Forrester and Senge, 1980; Sargent, 1991, 1996]. A general methodology for such a V&V process is developed later in this chapter.

Reliable data is at the heart of reliable models. Many of the validation techniques discussed with respect to model structure in subsequent sections are directly applicable to data as well. This research does not, however, focus on V&V for data specifically. However, the type, fidelity, or reliability of good data for use within the simulation often

drives the assumptions used in developing the model. Therefore, the modeler should make sufficient efforts to validate the data used in the model.

3.2 Taxonomy

There are a wide variety of techniques that can comprise a methodology for building confidence in the results of a model. The level of confidence needed in a model will vary as well, depending on the intended application and the risk associated with using incorrect model results. Different techniques inspire confidence at different levels of formality and rigor. The literature classifying the techniques based on the application of the V&V techniques is lacking. This section presents an original taxonomy of verification and validation techniques based on the function of the method (whether verification or validation) and the type of confidence inspired.

Two generalized verification and validation taxonomies were found in the literature [Davis, 1992; Balci, 1994]. However, important deficiencies were found in each. First, these taxonomies were developed prior to the recent explosion of interest in agent-based modeling. Not surprisingly, neither covers these types of simulations. Second, in Davis (1992), the presented taxonomy lacks basic definitions with respect to the V&V categories making its use somewhat arbitrary, and defining methods of V&V within its context difficult. Third, the general V&V taxonomy presented in [Balci, 1994] is identical to the verification (only) taxonomy in [Whitner and Balci, 1989]. Though V&V are important as a holistic process, each has a distinct function, and the tools associated with each are quite distinct [Caughlin, 2000]. Therefore, a taxonomy should

acknowledge the difference in functionality and purpose between verification and validation.

Figure 3.1 depicts a graphical representation of the V&V taxonomy based on the intended focus of the technique. The taxonomy is based on three general classification categories each for verification and validation. Each of the six are defined and illustrated with examples in the following sections.

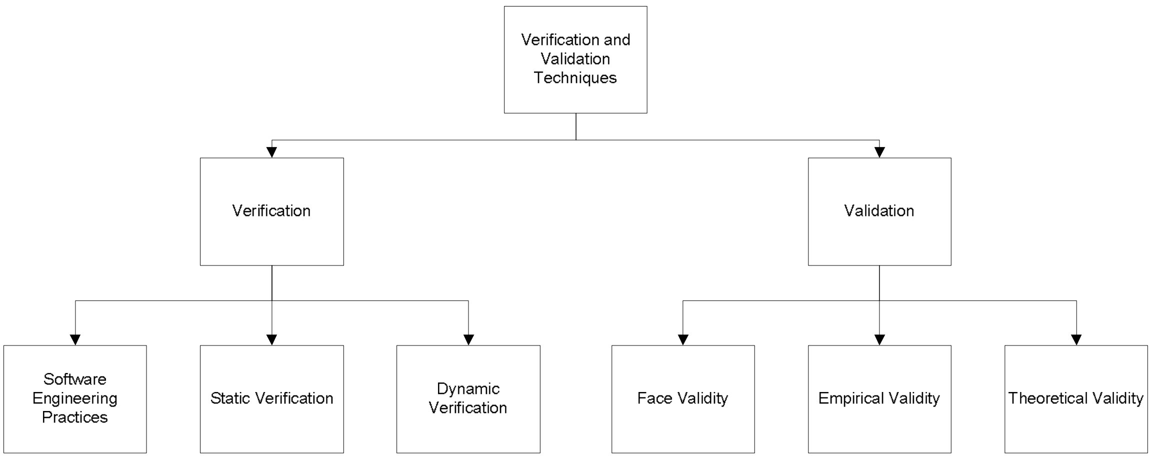

Figure 3.1 Verification and Validation Taxonomy

3.2.1 Verification Classifications

There are three general approaches to ensuring an executable model accurately represents the conceptual model. These verification categories are: software engineering practices, static verification, and dynamic verification. These categories represent a natural classification based on verification techniques used in constructing the model, to check its implementation prior to execution, and to check its correctness when running under various conditions, respectively.

3.2.1.1 Software Engineering Practices

The field of software engineering has produced a number of practices that aid the verification process by reducing the number of potential areas of error. Most modern computer programming languages actually require adherence to at least some of these practices. Some common examples of good software engineering practices applicable to the modeling process are described below.

<u>Logical and data flowcharting</u>: Logical and data flowcharts express the conceptual model in terms of an algorithm and data requirements. These charts reveal the structural and data requirements of the model, enabling a faithful translation of the conceptual model into an executable form. Once the model is built, logical and data flowcharts become a powerful tool for both static and dynamic verification techniques.

<u>Strong variable typing</u>: Variable typing is the method computer programming languages use to determine the amount of memory required to internally store the values assigned to the variables during execution. Each variable type is capable of storing its data to a specific precision. Strong variable typing does not allow data of a greater precision to be stored in a variable typed as having a lesser precision. This prevents unintended loss of precision during program execution.

<u>Modular design</u>: Modular design is a method of program coding that groups program statements according to some common functionality. In its basic form, sub-modules (also called functions or procedures – depending on the actual programming language used) are formed, allowing utilization of a single segment of code from multiple

other places within the code. Therefore, instead of multiple areas of identical code having potential errors, verification efforts can be concentrated on a single sub-module.

Object-Oriented design is modular design taken to an extreme, with all computer code encapsulated in modules called objects. Agent-based programming is an extension of OO design that groups all functionality associated with an entity into a single object. Many of the aims of OO design support ease of model verification by stressing the reuse of previously verified objects [Sycara, 1998].

Extensive documentation: Documentation, both internal and external, allows programmers, maintainers, and third party auditors to determine easily the intent of the documented code and, as a result, to identify coding logic that does not conform to the conceptual model. Documentation also facilitates many static verification methods.

Built-in error identification: A particularly effective verification method is to program checks into the model at data entry points. Also known as "trapping" or "handling," this technique allows the programmer to install verification into the model itself. When used in conjunction with dynamic verification methods, error trapping can be a powerful tool in identifying and isolating "spurious logic" [Davis, 1992]. An example of this technique is defining a specific range constraint for a variable.

Automated code generation techniques: As the OO and agent-based paradigms grow in popularity, there are an increasing number of development environments that allow the programmer to define, graphically or by some markup syntax, the structure of the object. In many cases, the environment generates the code necessary to implement the specified structure. Automated code is less prone to syntactical errors that must be

identified and corrected, enhancing ease of model verification. These techniques provide the added benefits of requiring object (agent) structures prior to coding and providing structural representations that can be used in common static verification techniques.

3.2.1.2 Static Verification

Static verification techniques are techniques implemented prior to running the model used to ensure accuracy in the executable model. These methods are concerned with the implemented accuracy of the model source code. As computer programming environments become more sophisticated, many of these methods have been automated [Balci, 1994], and current language compilers perform verification activities that fall into the static verification classification.

Code “walkthrough”: Code “walkthrough” encompasses a number of techniques used to verify the accuracy of programming code before execution. The techniques range from the informal *desk-checking* [Whitner and Balci, 1989], where the programmer steps through the code, to a *structured walkthrough* process [Sargent, 1991, 1996; Balci, 1994], a formal process involving a review team charged with evaluating the model relative to specifications and standards and reporting deficiencies.

Structural verification test: Structural verification tests ensure the structure of the model does not contradict knowledge about the structure of the conceptualized system. During structural verification, data and logical flowcharts can be compared to the structure of the executable model to help identify structural deficiencies in the model implementation. These tests also are used to identify and verify assumptions are correctly implemented [Forrester and Senge, 1980].

In the case of OO or agent designed models, the object structures can be compared to the conceptual entities. Multi-agent modeling verification techniques in this category include verification of communication states and protocols that prevent deadlocks. Some agent development environments perform communication verification as a component of their automatic code generation functions.

Syntax checking: Syntax is the "grammar" that allows higher level programming languages to be translated into machine executable code. Most modern model development environments provide surface-level syntax checks as the code is typed. Compilers, the automated translators, perform additional syntax checks and provide a host of structural information when generating the executable model that can be used to verify variable declarations, modular structure, and sub-model interfaces [Whitner and Balci, 1989].

3.2.1.3 Dynamic Verification

Dynamic verification techniques are those that require the execution of the model and test model correctness under run-time conditions. These techniques entail gathering observations of executing system behavior. Some dynamic verification techniques are aided by automated tools available in model development environments. More so than with static verification, dynamic verification relies more on the model programmer to develop and implement the tools used to evaluate the correctness of the executable model. Examples of some of the more common techniques are found below.

Model instrumentation: Model instrumentation is the technique whereby the modeler builds verification cues into the execution code to provide data necessary for

verification. As the model executes, the instrumentation code collects information and reports on the system states, both model and program. This information is then used to determine model accuracy.

Most modern development environments go a step further by providing automated instrumentation aids through a runtime debug mode. Typically, the runtime debug mode provides for line-by-line execution of the model, which allows watches to be set on different variables, execution breaks (or pauses) at desired points in the execution, access to stack contents (representing sub-module call orders), and other execution state information.

<u>Testing based on model development strategies</u>: There are two purist approaches to testing, top-down and bottom-up. The actual choice is based on the model development strategy used. In *top-down* development, model construction begins with the sub-models at the highest level and ends with the sub-models at the base level. Conversely, bottom-up development begins with the construction of the base models, models where no more decomposition is possible or desirable, and ends with the integration of all sub-models to form the top level model.

Top-down testing begins by testing the model at the highest level. Calls to lower level sub-models are simulated (also known as "stubbed out"). As each sub-model is developed and tested, it is added to the global model and the global model is again subjected to testing. The process continues until the base level models have been integrated into the global model.

Balci (1994) notes top-down testing has advantages and disadvantages. Advantages include: early existence of a working model; the top level model becomes a natural environment for testing lower level sub-models; and errors are localized to newly added sub-models. Disadvantages, however, arise from the fact that testing can only occur by running the entire model. This results in discouraging thorough testing of the sub-models and their integration.

Bottom-up testing begins by testing each sub-model thoroughly and when sub-models belonging to the same higher level model are completed, they are integrated and their integration tested. This continues until all sub-models are integrated forming the completed model.

Whitner and Balci (1989) note bottom-up testing has advantages and disadvantages. The primary advantage is a more thorough testing of sub-models, since sub-models typically represent less complex functions than their aggregates. The main disadvantage is that sub-model testing requires individual drivers, or harnesses, for each sub-model, and the development of separate drivers can be quite expensive.

Sargent (1996) writes that bottom-up and top-down testing can be combined to conduct *mixed testing*.

<u>Path Analysis</u>: Path analysis attempts to identify the possible state paths the model can take and, by generating appropriate input data, to force the model along each path. Complete path testing not only ensures each path can be reached, but also checks that paths are properly taken with intended values.

Boundary analysis: Boundary analysis methods are used to check model behavior at and near threshold values. These thresholds are values at which system state changes take place, as well as along variable limits. This technique is used in deference to the fact that errors lie along boundaries [Whitner and Balci, 1989; Balci, 1994].

Execution monitoring: Execution monitoring encompasses a variety of techniques used to provide a description of the model's activities during execution. Three such techniques are tracing, visualization, and assertion checking. *Tracing* is automatically getting all intermediate results during program execution [Kleijnen, 1995a]. The trace, the recorded log of the intermediate results, is analyzed to determine whether or not the program is functioning correctly (as intended). *Visualization*, or *animation*, provides for visual inspection of the modeled system during execution, which can highlight unintended system behaviors. *Assertion checking* internally monitors system states or specifications and reports when the simulated system violates intended limits.

3.2.2 Validation classifications

Validation determines how accurately a model represents the real-world system. There are three broad approaches to model validation: face validity, empirical validity, and theoretical validity. These categories broadly represent the majority of validation techniques available using experts, observed data, and scientific theory. Just as all validation techniques may not be applicable to every model, a given application may not need to achieve each facet of validity [Davis, 1992]. The mix of techniques (and ultimately the amount) used for any particular application of the model is a function of the level of acceptable risk involved in using the model.

3.2.2.1 Face Validity

Techniques used to establish a level of face validity are primarily concerned with providing confidence that, on the surface, the model appears reasonable to those knowledgeable about the real-world system [Law and Kelton, 1991]. Techniques in this category range from "eyeballing" [Davis, 1992] to formal Turing tests. This approach to validity is based on the notion of a rationalism approach to model validation [Naylor and Finger, 1967].

Rationalism [Naylor and Finger, 1967]: The conceptual model, developed through study of the system and conversations with the system experts, is reduced to a set of postulates. These are then presented to the experts for refutation or adjustment. When these postulates are sufficiently rigorous in the judgment of the experts, then the resulting model has high face validity. It is supposed that with accurate translation into the executable model, that model too will have high face validity.

Graph-based analysis: Graph-based analysis brings together many components used in other verification and validation to establish the face validity of the model. Within the context of a formal walkthrough, graphical representations of the conceptual model, including system and entity structure, are presented to the system experts for review.

Prototyping: In prototyping, a rough, first-cut executable model is produced and evaluated for basic behavior. The intent is to validate the conceptual model and to identify significant areas that were neglected in its formulation. In addition, the prototype can be used for initial sensitivity analyses and to identify significant parameters

affecting system behavior in the model. These prototypes are sometimes built in a language specific for prototypes. This means later re-coding of the prototype into a production language.

Animation: Though animation is also classified as a verification technique, it can be a powerful tool in helping to build face validity. Instead of looking for unintended system behavior (verification), system experts review the model's behavior to determine if it is representative of the real-world system. If the behavior is not representative of the real-world system, the experts can help in the identification of conceptual errors that led to the questionable behavior [Kleijnen, 1995a]. A key assumption, of course, is that the animation-to-model linkage has been verified and is thus accurate.

Turing test: A formalized Turing test [Russell and Norvig, 1995; Balci, 1994, Kleijnen, 1995a] involves mixing a number of real-world system performance indicators with those produced by the simulation. System experts are then asked to identify which are from the real-world system and which are from the simulation. The less the experts can distinguish correctly between the outputs, the higher the degree of validity in the model.

The Turing test does require real-world data. If the simulation is of a non-existent or purely theoretical system, then there may not be real-world data for comparison. For example, in the case of modeling combat or other systems where costs are extremely high in time, money, or life, there may be some real-world system data, but it may be too scant for sufficient Turing tests.

Documentation: When the model code is documented to demonstrate data sources, assumptions, and component validation results, it becomes a valuable tool for establishing face validity. When the model is subjected to third party, or independent, validation, this type of documentation is critical.

3.2.2.2 Empirical Validity

Given that the purpose of a model is to represent a complex, real-world system, the aim of empirical validity techniques is to provide an indication as to the accuracy of the model with respect to the observed behavior of the system under study. These techniques are used to establish a scientific basis for confidence, but they stop short of offering absolute proof that the model results are an accurate representation of the real-world system.

Statistical Techniques: Statistical techniques are particularly useful when the system is observable (i.e. it is possible to collect a reasonable amount of data on its operational behavior [Sargent, 1996b]) and output data are used to compare model output with that of the real-world system under sufficiently similar configurations. There have been many statistical techniques proposed for use in validation of models (and sub-models). Balci (1994) presents a table of 18 different techniques and associated references.

Depending on the risk associated with the model, absolute accuracy may not be necessary. Some "weak" regression techniques have been proposed that indicate some appropriate correlation between the model and real-world system under similar inputs can be a valuable validation tool as well [Kleijnen, 1995a].

Graphical Validation: In cases where statistical tests are not appropriate because the assumptions cannot be satisfied, observations of the real system are too limited, or the output process is highly non-stationary, non-statistical comparison methods are available. Sargent (1996b) presents subjective, graphical methods of comparison including histograms, box plots, and behavior graphs.

Sub-model Validation: Sub-model validation provides a strong indication that the composite model is also valid. However, since errors are compounded in the aggregation of validated sub-models, it is not sufficient in and of itself. Multiple sub-models that produce acceptably accurate results may, when integrated with one another, produce system results outside acceptable bounds [Balci, 1994]. In spite of this complication, sub-model validation is an important component of building confidence in the overall model.

Historical or field test data: When the real-world system does not exist, comparison to field test or historical data is often possible. This data can give an indication of how the proposed system should (or did) behave, and a favorable comparison to the model behavior can build confidence in the model.

Comparison to other models: Comparing a new model to another well accepted (validated or not) model is another empirical validation technique. However, there are two issues that must be addressed. First, the success of this method depends in a large part to the degree the "old" model is deemed correct. Second, in the case that the "new" model is significantly better than the "old," the discrepancy may cause results from the

new model to be unjustly doubted. However, more confidence can be built when results from both models, established and new, agree.

3.2.2.3 Theoretical Validity

Theoretical validity encompasses the techniques used to establish the extent to which a model conforms to scientific theory. The techniques in this category are largely used to prove mathematically a model is correct. Balci (1994) notes that "current state-of-the-art formal proof of correctness techniques are simply not capable of being applied to even a reasonably complex simulation model." He goes on to list seven common proof of correctness techniques: induction, inference, λ-calculus, logical deduction, predicate calculus, predicate transformation, and proof of correctness.

Some of the theoretical validation techniques are finding applicability in agent-based models, particularly in the validation of single and multi-agent systems comprised of intelligent agents. Planning and problem solving functions are often based on predicate calculus and logical deduction. Theoretical validation techniques are being used to prove that the knowledge-based model is correct [Jabbar and Zaidi, 2001].

Additionally, theoretical validation of sub-models may be possible. For example, a sub-model calculating a shortest path may be proved mathematically correct. Theoretical validation of sub-models can be a significant step in the validation of the aggregate model, though, as before, it is not sufficient since other sub-models can introduce enough error to "invalidate" the combined model.

3.3 V&V Methodology

As previously indicated, early computer simulation researchers were aware that V&V should be an integral part of the modeling process. Mihram (1972) proposed a five step modeling process (adapted for Figure 3.2) that contains many of the basic components of modeling processes used today. Step 1, system analysis, involves defining the experiment, asserting the assumptions, and abstracting the system into a conceptual model. Step 2, system synthesis, is translating the conceptual model into an executable (computer) simulation. Step 3, verification, includes all techniques to ensure that the executable simulation is an accurate representation of the conceptual model. Step 4, validation, encapsulates all methods used to build user confidence that the model is an accurate representation of the real-world process or system. Finally, step 5, model analysis and inference, includes conducting the experiment and subsequent analysis necessary to support the purpose (intended application) specified for the model in step 1.

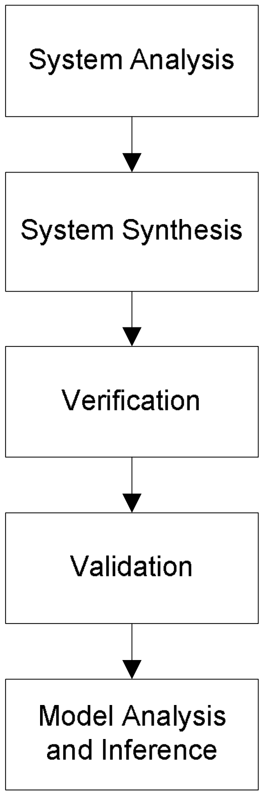

Figure 3.2 Modeling and Simulation Process

Though Figure 3.2 generally contains all steps in more recently proposed M&S processes, it does not acknowledge the iterative nature of M&S. Feedback from both verification and validation can be (and is) used to refine the conceptual and executable models to make the simulation more robust when experimentation and analyses are ultimately conducted. Recognizing that the V&V process is iterative, Law and Kelton (1991) proposed a simulation study process including feedback. The Law and Kelton process was generalized for the modeling and simulation process shown in Figure 3.3.

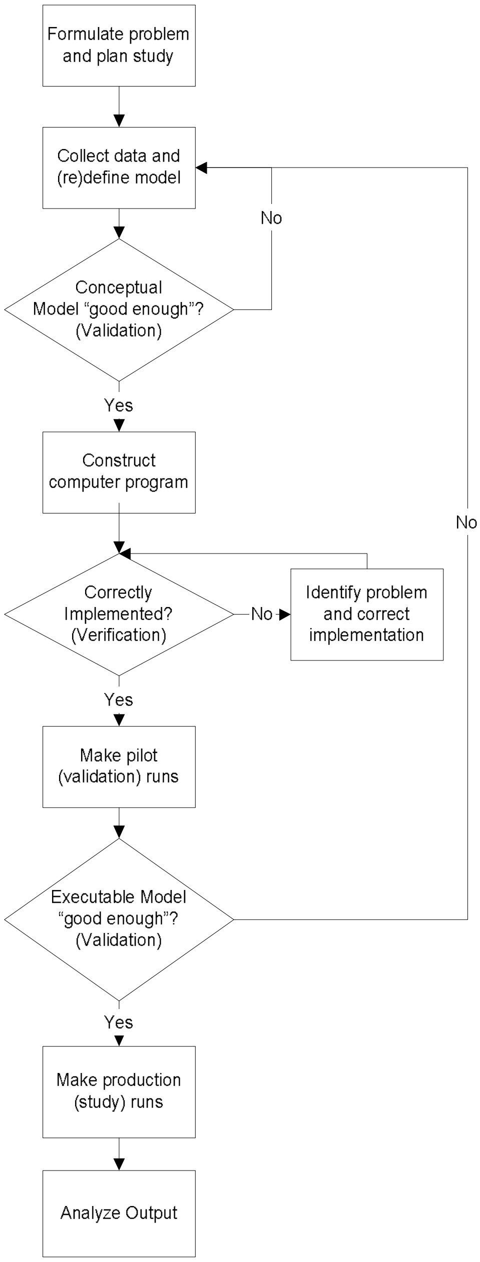

Figure 3.3 Generalized Modeling Process with Feedback

The Law and Kelton process defined in Figure 3.3 makes several key improvements to the process shown in Figure 3.2. First, there are three points of feedback that are used to improve the fidelity of the model under development: 1) after development of the conceptual model (validation); 2) after coding the executable model

(verification); and 3) after validation runs, but before the experimental runs (validation). Second, it recognizes the necessity of taking steps to validate the conceptual model before translating it into the simulation environment.

Sargent (1996a) presents a more compact modeling process (Figure 3.4). In Figure 3.4, the modeling process begins with the "Problem Entity" box and moves clockwise as the modeling process progresses. This representation of the modeling process is particularly useful in that it depicts the V&V activities (outside, solid arcs) occurring in conjunction with the model development, coding, and experimentation (dotted lines connecting the modeling objects). This view is more consistent with the V&V literature, which stresses ongoing and continuous V&V throughout the lifecycle of a model [Law and Kelton, 1991; Balci, 1994; Sargent, 1996a; Nayani and Mollaghasemi, 1998]. Additionally, it links the data, central to modeling fidelity, with the overall modeling process.

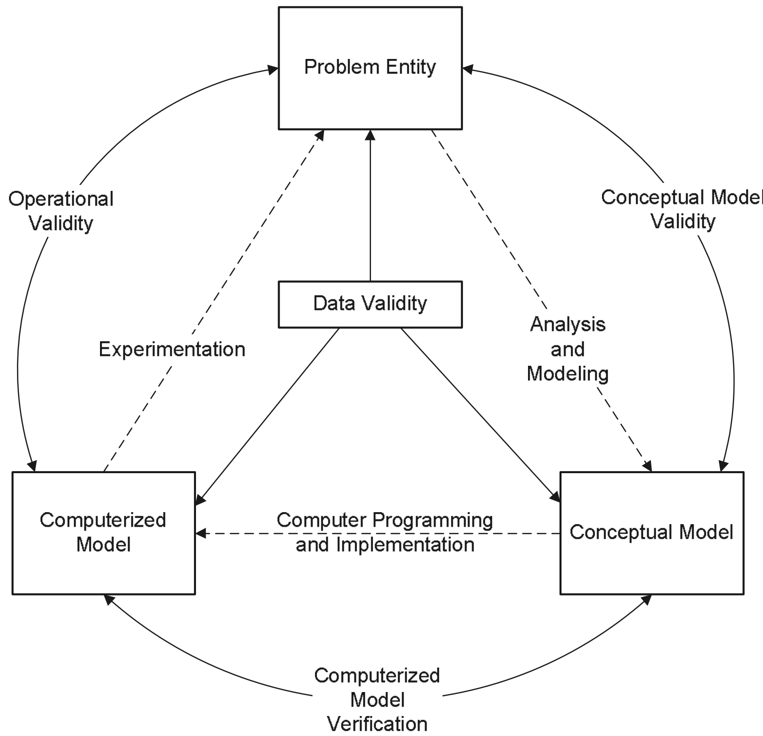

Figure 3.4 Simplified Modeling Process (Sargent, 1996a)

Despite the various models indicating where V&V efforts belong in the modeling process, there is neither a common template indicating which techniques should be used, nor is there commonality or agreement indicating how much V&V is ultimately necessary. Instead, it is left to the organization and/or individual employing the simulation to determine the methods and extent of V&V efforts needed to inspire sufficient confidence in the simulation results.

Chapter VI presents a case study of the verification and validation process developed in this research generally following the modeling process found in Figure 3.3, but expanded to include conceptual model feedback from both the verification and executable model validation processes as indicated in Figure 3.4.

3.4 Does V&V Ultimately Matter?

V&V is quite important. However, a model that has not been validated may not be useless. There are cases when a model cannot be validated against any knowable data, experimentation is too costly (by some measure: cost, lives, risk, etc.), cases when only the conceptual model and/or sub-models can be validated, or the model is the best-known (best-guess) representation of the real-world system (e.g. campaign-level models of combat). In these cases, attempts to V&V the completed model may be incomplete at best, but at the same time, the model may be necessary. Hodges (1991) argues that in these cases, the models can be useful evaluation tools, even though their predictive power is suspect [see also Hodges and Dewar, 1991].

IV. Bay of Biscay Agent-Based Simulation

Agent-based combat simulations to date have been relegated to small, toy scenarios with sometimes tenuous links to real-world operations. As a result, little can be said about the true degree to which agent-based models are applicable to solving real-world military problems. This chapter addresses this void by describing an agent-based combat simulation built around an historical example of offensive search. The result is a first-ever agent-based mission-level model demonstrating a significant level of validity (detailed in Section 4.4) and potential applicability to a wide range of modern scenarios, including military, law enforcement, immigration, and international treaty verification.

The real-world operation selected for the simulation application was the offensive search for U-Boats in the Bay of Biscay by the Allied forces during World War II. This chapter provides a brief historical description of the Allied operation, details the assumptions and implementation of the computer model, and applicability of the simulated scenario to modern military and domestic security problems.

4.1 The Historical Operation

German U-Boats operated against Allied shipping in the North Atlantic from 1941 through the end of the war in an effort to reduce the shipments of war-time supplies to Great Britain. Following the fall of France, many of these submarines operated from ports in occupied France, crossing the Bay of Biscay into the North Atlantic, where they hunted for Allied transport ships. Once they left the Bay of Biscay, the U-Boats could

operate outside the reach of Allied aircraft support. For a time in 1942 and 1943, this offensive was so successful that Great Britain's war effort was put in great peril.

While the Allied forces had little hope of finding and destroying U-Boats once they reached the Atlantic, the Bay of Biscay was well within the reach of Allied aircraft. Additionally, the amount of U-Boat traffic to and from the French ports, necessitated by maintenance and resupply/refuel demands, ultimately meant that there was sufficient density of targets within the Bay of Biscay to warrant committing resources to conduct anti-U-Boat efforts. As a result, the Allied forces, beginning in 1941, hunted for the U-Boats in the Bay of Biscay.

Both the Allies and the Germans were able to consistently add technological advances to their forces during these U-Boat operations. Additionally, as each side was able to identify their opponent's new advance, they were able to modify their own tactics or improve upon existing countermeasures to eventually mitigate the innovation. As a result, the "measure-countermeasure" seesaw of technology and tactics is prominent throughout the operations.

Additional historical background on the offensive search in the Bay of Biscay can be found in [McCue, 1990], and an extensive record of the corresponding operational analyses may be found in [Waddington, 1973] and [Morse and Kimball, 1998].

4.2 Model Description

The Bay of Biscay agent-based simulation was built to reproduce the results of the historical operation in both qualitative and quantitative measures. A development goal

was to keep the simulation relatively simple by including only the most significant factors and to make explicit use of agents. As a result, assumptions were made regarding the simulated system.

4.2.1 Assumptions

Constructing the Bay of Biscay agent-based simulation required assumptions about the environment, the aircraft agents, and the U-Boat agents. The following sections detail the primary assumptions made to represent operations and tactics from both the Allied and German perspectives as faithfully as possible without including an inordinate level of detail.

4.2.1.1 Environment

<u>Daylight</u>: Both U-Boat surfacing policy and aircraft effectiveness were governed by day versus night conditions. Within the simulation, "day" is defined as the time between nautical dawn and nautical dusk (i.e. sun is above -12º with respect to the horizon). Daylight computations are approximations made with respect to a single point near the geographical center of the Bay of Biscay and applied to all locations in the simulation. Since daylight times do not differ significantly within the area encompassed by the simulation, the single point calculation does not introduce an unreasonable amount of "daylight" error. In fact, in [McCue, 1990], daytime calculations failed to include dawn and twilight times, which resulted in underestimation of the amount of daylight by as much as 30-60 minutes of light daily [McCue, 2002].

Sensors: All detection sensors assume conformity to the Inverse Cube Law. The Inverse Cube Law states that the probability of detection is inversely proportional to the cube of the distance between sensor and target. This assumption is supported by field testing performed during WW II [McCue, 1990; Waddington, 1973; Morse and Kimball, 1998].

The Inverse Cube Law is an important assumption as it provides a convenient closed-form solution for combinations of conforming detection sensors. When more than one sensor is used, the resulting sweep width, or effective sensor range, is approximated as the square root of the sum of squared sweep widths for the individual sensors (4.1). Specific sweep widths for independent sensors were obtained from [McCue, 1990].

$$W_{total} = \sqrt{\sum_{i=1}^{n} W_i^2} \tag{4.1}$$

where W_i is the sweep width of the i^{th} sensor
n is the number of independent sensors.

There are two issues important to independent sensor combination calculations. First, the approximation breaks down when the number of independent sensors, n, is increased sufficiently. For example, no combination of sensors would allow for a positive probability of detection for objects beyond the horizon. Second, the probability of detection, given by (4.2) [McCue, 1990], provides for positive probability of detection regardless of the distance between the sensor platform and the target.

$$P(x) = 1 - e^{\left(\frac{W^2}{4\pi \cdot x^2}\right)} \tag{4.2}$$

where W is the sweep width computed by (4.1), and
x is the distance of target-sensor separation.

Neither of the two issues above are factors in this simulation. The number of independent sensors is kept quite low ($n \leq 3$), which is sufficiently small to avoid an improbably large combined sweep width. A random detection check is made only when a target is within the sweep width of the sensor platform ($x \leq W$, (4.2)) to avoid making nonsensical probability checks when the target is impossibly distant from the searcher. This leaves a certain (minor) amount of detection probability unaccounted for, but the savings in computation time gained, as well as avoiding nonsensical detections, warranted this sacrifice in accuracy.

No-Fly Zone: The French ports used to base the U-Boats were heavily defended and protected by German air patrols. Additionally, U-Boats leaving and entering port areas had air escorts available to them. Therefore, simulation bombers generally standoff 100 NM from the coast of France in acknowledgement of this threat. Likewise, U-Boats take advantage of the escorts by running entirely on the surface once they move within 100 NM of the coast. More specific behaviors regarding the region of the bay within 100 NM of the coast of France are found in the following two sections.

4.2.1.2 U-Boat Assumptions

Information governing the German U-Boat tactics, policies, and operation was significantly more difficult to assimilate into the simulation than for the Allied agents. This was primarily due to conflicting information between available sources. In cases of conflicting information, especially between non-German sources, the source having the latest date of original publication was used, since typically the later studies had access to more declassified sources, both German and Allied.

U-Boat agents within the simulation must spend a minimum of 3 hours surfaced for each 100 nautical miles (NM) traveled to fully recharge their batteries. This is required because U-Boats involved in the Bay of Biscay operation were not outfitted with the snorkel, developed very late in the war, which would allow them to operate with their diesel engines while submerged. Therefore, within the simulation, all U-Boat agents simulate battery operation while submerged and diesel operation while surfaced. Upon battery depletion, the U-Boat agent would coordinate the timing of its surfacing to coincide with its surfacing policy (i.e. day or night). Both battery charge and discharge is assumed to be linear with respect to time surfaced or distance traveled while submerged, respectively.

U-Boats traveled to and from port via an essentially East-West trajectory within the Bay of Biscay [McCue, 1990; Waddington, 1973]. U-Boat movement is 10 knots (NM/hour) surfaced and 2.5 knots submerged.

U-Boat agents leave port with thirty days of supplies and time their return from operations in the North Atlantic to arrive back in port with no supplies remaining. Additionally, the effect of limited U-Boat refueling at sea is implicitly modeled by allowing a 0.25 probability of extending their time in the North Atlantic by 30 days. This fraction of the operational fleet also included a common practice of commanders extending their operational tour to 60 days by stretching their initial resources [McCue, 1990; Morse and Kimball, 1998].

Throughout the war, anti-aircraft artillery from the U-Boats was ineffective. Therefore, it was generally German policy to submerge when Allied aircraft was sighted.

Therefore, U-Boat agents in the simulation submerge immediately upon detecting an aircraft, regardless of their battery recharge state. Once submerged, these agents will travel submerged until their battery level is depleted and coordinate the timing of their surfacing to coincide with the fleet's surfacing policy.

Regardless of surfacing policy, the U-Boats in the simulation operated in a surfaced state while they were in the 100 NM coastal region protected by German air patrols.

Perhaps the biggest unknown factor regarding actual U-Boat activity concerned the time spent in port, and this remains the biggest unknown regarding the link between the Bay of Biscay agent-based simulation and the real-world operation. There was simply not enough data available to support anything but reasonable assumptions. In the simulation, U-Boat time in port is modeled as a uniform random variable between 25-40 days, inclusive. This is derived from [Morse and Kimball, 1998] which states that the U-Boat would spend "about 30 days" in a port operating under its capacity (no strict queuing argument is attached to the word capacity in this instance). However, from other sources, most notably [McCue, 1990], the French ports were often choked beyond their ability to service all the boats, especially toward the end of the war when German resources became scarce.

4.2.1.3 Aircraft Assumptions

Over the Bay of Biscay, Allied aircraft operated with impunity, since German U-Boats had ineffective active defenses (i.e. anti-aircraft artillery) and the search area was outside the range of German fighter escorts. While there were undoubtedly accidents

involving the loss of aircraft over the length of the campaign, the offensive search for U-Boats constituted a small force of aircraft, and the available fleet used for this purpose was not impacted by such occurrences. As a result, there is no attrition due to accident or anti-aircraft defenses modeled within the simulation.

The simulated Allied aircraft agents standoff from the coast of France to avoid enemy air patrols and escorts. Agents generally do not enter the 100 NM coastal no-fly zone region. The one exception is the case that an aircraft locates a U-Boat prior to the U-Boat entering this region. In this case, the aircraft follows the U-Boat into the region to attack it. Following the attack, the aircraft immediately exits the hostile region.

Aircraft agents move at a constant speed of 120 knots, and the effects of weather once a mission is launched are not simulated. Once airborne, each aircraft flies up to 70% of its fuel load, or until it has expended its munitions. This fuel factor is supported by subsequent analyses [Waddington, 1973] in spite of policy indicating pilots were to fly up to 80% of their initial fuel capacity.

Simulated aircraft can detect only surfaced U-Boats. Once spotted, an aircraft pursues the U-Boat until the attack is made, to the exclusion of all other considerations. In attacking a U-Boat, the aircraft agent expends its entire payload of munitions and returns immediately to its base.

Weather and maintenance problems were a big issue with respect to successful Allied operations, and each factor is modeled stochastically. At the beginning of each simulated day, a random draw is made to determine if the weather grounds the entire fleet for that day. Maintenance, on the other hand, affects aircraft agents individually and is

determined immediately prior to take-off. Once in the air, the aircraft agents do not abort due to poor weather or maintenance problems. Aircraft return to base only for fuel or munitions.

4.2.2 Conceptual Models

In building the Bay of Biscay agent-based simulation, the scenario was decomposed into two separate processes, U-Boat Flow and Aircraft Flow. Each process models the operational and support elements of the respective forces.

Figure 4.1, adapted from [McCue, 1990], illustrates the basic conceptual processes influencing the flow of the U-Boats to and from their operating zone in the North Atlantic. U-Boats are individually assigned to one of five French ports and enter the Bay of Biscay *en route* to their operation zone in the North Atlantic. The U-Boats exit the Bay of Biscay when they reach the North Atlantic. Operations in the North Atlantic, to include refueling, are not explicitly modeled. Instead, the U-Boats remain outside of the Bay of Biscay for a length of time proportional to the amount of provisions remaining when they initially exit the bay. Refueling is implicitly modeled by a fraction of U-Boats extending beyond their initial provisions by an additional thirty days. When the provisions remaining reach a critical level, U-Boats re-enter the bay *en route* to their assigned port facility. Additional U-Boats enter the simulation from the German shipyards according to historical rates specific for the given time period being simulated, arriving in the North Atlantic with 30 days of provisions. U-Boats leave the simulation when sunk by Allied aircraft in the Bay of Biscay. The simulation does not account for U-Boats sunk during operations in the North Atlantic.

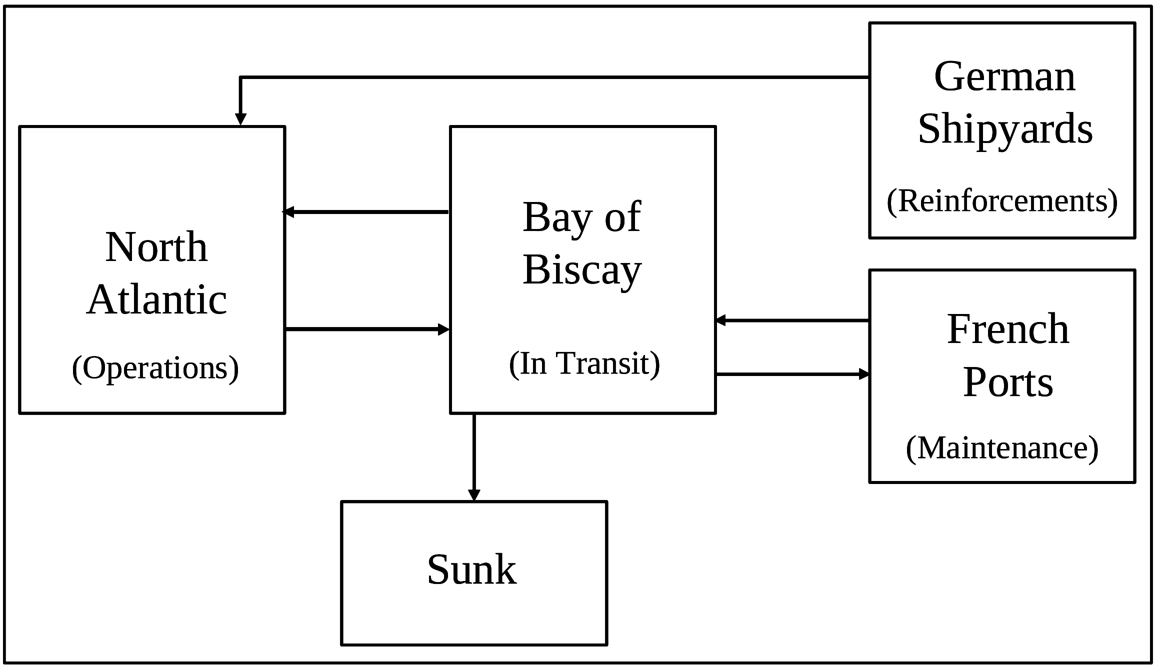

Figure 4.1 U-Boat Flow, Conceptual Model

Figure 4.2 illustrates the influencing processes of conducting offensive search in the Bay of Biscay by Allied aircraft. This model is significantly simpler than the previous agent flow model. Aircraft are assigned to a single base, enter the Bay of Biscay to perform their search, and egress when fuel reaches a critical level or their munitions are expended.

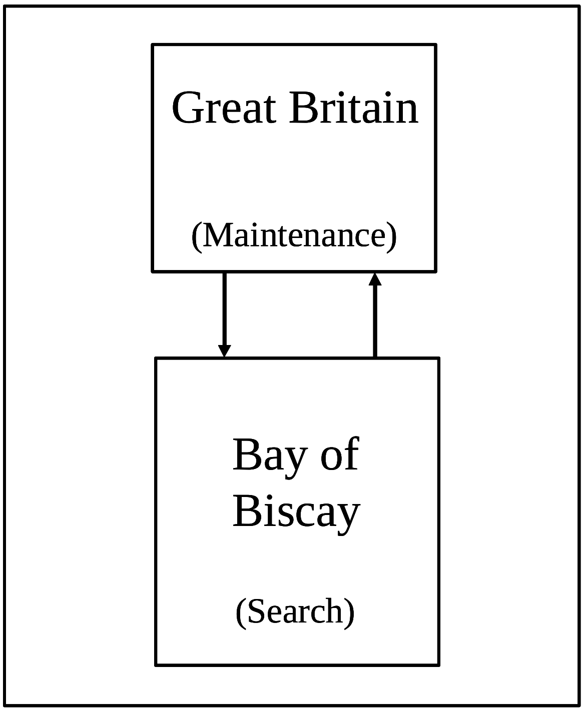

Figure 4.2 Aircraft Flow, Conceptual Model

The two conceptual models provide for interactions between the two agent types, which occur only over the Bay of Biscay.

4.2.3 Conceptual Model Validation

Before developing the executable model (code), a formal conceptual model was developed. Several techniques were used to establish the validity of this conceptual model, and these are discussed below.

4.2.3.1 Validation against previously validated models.

In the years following WW II, several mathematical models have been developed to analyze the anti-U-Boat operations in the Bay of Biscay. McCue (1990) details his model and presents a graphical depiction of his conceptual model of the U-Boat flow through the Bay of Biscay. The model elements are consistent between the two models.

That is, the conceptual model depicted in Figure 4.1 agrees with McCue's U-Boat circulation model with only minor differences in the level of fidelity for U-Boat operations within the North Atlantic.

There are two differences between the proposed Bay of Biscay model and McCue's U-Boat circulation model. First, unlike McCue's model, the U-Boat flow model of Figure 4.1 does not account for U-Boats sunk in the North Atlantic during their operational tour. Second, McCue's model explicitly allows for multiple refueling opportunities for U-Boats in the North Atlantic, while the model of Figure 4.1 does not. Instead, U-Boats in the proposed model are given a single opportunity to extend their operational time by 30 days according to historical figures.

The differences in the U-Boat models were not deemed significant for several reasons. First, the differences outlined above are the result of a slight difference of focus for the two models. While, the proposed model concentrates on measures of effectiveness (MOEs) within the Bay of Biscay, McCue's model was intended to provide additional insight into the effect on Allied transports in the North Atlantic as well. Therefore, additional fidelity in his model is more important to his measures. Second, McCue's model was intended to model the entire 4 year conflict, while the proposed model was built with a much shorter (6 month) time frame. The shortened time frame makes U-Boats sunk in the North Atlantic a less significant factor. This is due to the fact that U-Boats were much more likely to be sunk in the bay than in the North Atlantic [McCue, 1990].

4.2.3.2 Prototyping and Subject Matter Experts.

Following the development of the conceptual models, a prototype was developed and presented to subject matter experts [McCue, 2002], at the Center for Naval Analysis (CNA) in order to refine the conceptual models. Review by the subject matter experts suggested inclusion of the U-Boat reinforcement component in Figure 4.1, which was born out by subsequent output analysis. Additionally, implementation of the models was modified to prevent the Allied aircraft from flying over the occupied French territories.

4.2.3.3 Preliminary Output Analysis

In addition to the subject matter expert review, preliminary output analysis suggested that the reinforcement component of Figure 4.1 was needed, and there were two indications for this. First, without German reinforcements, the number of U-Boat sightings trended down during the simulation as the German fleet was attrited. Simulating the reinforcement process according to the historical numbers alleviated this problem. Second, without the reinforcements, the U-Boat arrivals into the bay were not distributed Poisson, as were the historical arrivals. The arrival process with reinforcements was much closer to Poisson distributed (see section 4.4.4).

4.2.4 Conceptual Model Implementation

The Bay of Biscay agent-based simulation was written in JAVA® (version 1.4.1) and executed on a 2-GHz Pentium 4® PC with 256 MB of RAM running a Windows® 2000 operating system. The simulation is comprised of 37 classes (objects) with more than 10,000 lines of code including internal documentation. The simulation used between 3 and 6 seconds elapsed time per simulation day, depending on the number of

agents active (i.e. in or over the Bay). Within the simulation, each U-Boat and aircraft is an agent running in an independent thread of execution, with additional threads for the GUI controls.

The simulation itself was written to operate in any of three modes. The first two modes allow for demonstration and model verification (debugging). One provides for running through the operating system (command prompt), and the second provides for running the simulation through a JAVA capable web browser. Replications are not possible when running in either of these modes, and therefore, no statistics are kept. The third mode of operation, called batch mode, provides a method of running a user-specified number of replications, and statistics are kept on a number of measures of effectiveness (MOE). Batch mode is the only mode appropriate for practical quantitative analyses.

Agent and simulation design data was compiled according to the following hierarchy: 1) historical fact as found directly from sources credited to Allied and German participants; 2) published studies directly related to the offensive search in the bay; 3) data derived from raw numbers in one or more of the preceding sources; and 4) good judgment (operational expertise) when the three previous sources fail or contradict one another.

4.2.4.1 Agent Decisions and Movement

The agent environment was discretized into a 800 x 680 pixel grid, with each pixel representing about 0.9024 NM for a total of just under 391,000 NM^2 of territory simulated. Each agent is capable of traveling a specific distance (STEPSIZE) based on

speed and simulation time elapsed since its last move. This provides a grid of discrete locations to which an agent can move during an update (see Figure 4.3).

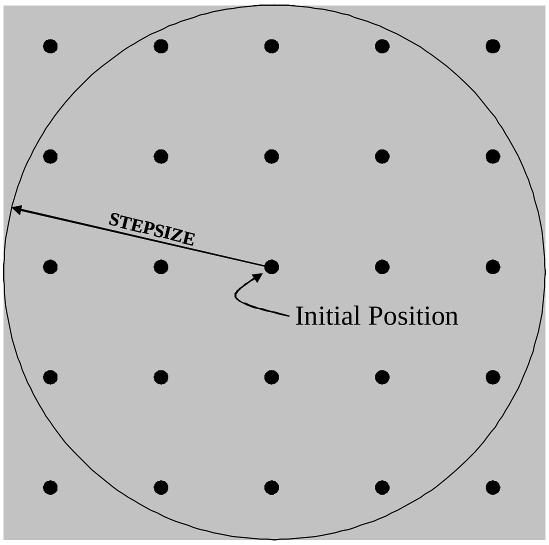

Figure 4.3 Possible Agent Moves

All grid points (nodes) within the circle (of radius STEPSIZE) are reachable in the next possible move. The agents choose between the possible nodes by evaluating a penalty function and selecting the node with the minimum penalty. Aircraft and U-Boat agents utilized different penalty functions within a decision hierarchy particular to each agent type, aircraft or U-Boat.

4.2.4.1.1 U-Boat Behavior

U-Boat agent behavior is determined through a hierarchical decision process based on its current state. A U-Boat agent makes behavioral decisions according to the hierarchical priorities listed below:

U-Boat Agent Decision Hierarchy

1. Avoid contact with Allied aircraft (surfaced U-Boats only)
2. Battery state
3. Conform to surfacing policy
4. Move

Avoid contact with Allied aircraft. The foremost priority for a U-Boat on the surface is to avoid contact with the Allied aircraft searching the Bay of Biscay. Each surfaced U-Boat attempts to detect any aircraft within its combined sensor range. If an aircraft is detected, the U-Boat submerges. Otherwise, the decision falls to the second tier of the hierarchy.

Battery State. If the U-Boat does not detect aircraft within its combined sensor range or it is submerged, then the state of the battery charge is the next factor in determining its actions. If the U-Boat is on the surface and the batteries are fully recharged, then the U-Boat is prepared to submerge. If, on the other hand, the U-Boat is submerged and the batteries are depleted, then the U-Boat is prepared to surface. Given these two conditions, the decision to change submergence states falls to the third tier of the decision hierarchy. In the absence of either of these conditions, the U-Boat maintains its current battery state (i.e. charging on the surface or depleting while submerged), and the decision falls to the fourth tier (Move).

Conform to surfacing policy. The third tier of the decision hierarchy ensures that the surfacing policies are enforced. If surfacing or submergence criteria are met, then the U-Boat chooses to change its submergence state to the desired state value. Otherwise,

this tier forces the U-Boat to maintain its current state until the policy criteria are fulfilled.

Move. The fourth level of the decision hierarchy determines the coordinates the aircraft agent moves to during the current agent update. The move coordinates are selected via a penalty function evaluation. The penalty for moving to some proposed coordinates (i, j) is comprised of four component penalties (k = 1, 2, 3, 4). For k = 1, the penalty component is computed as the 2-dimensional Euclidean distance between the proposed move location (i, j) and the ultimate goal coordinates (x_{goal}, y_{goal}):

$$P_{i,j}^{<k>} = \sqrt{(x_{goal} - i)^2 + (y_{goal} - j)^2} \text{ for } k = 1. \tag{4.3}$$

The remaining penalty components represent environmental knowledge of past interactions (events) between the opposing forces. The event-based penalties (*k* = 2, 3, 4) have the same form given by:

$$P_{i,j}^{<k>} = \sum_{\substack{type-k \\ events}} A_k \cdot e^{\frac{2 \cdot \ln(0.5) \cdot d}{r}} \quad \forall\, d \le r,\, k = 2, 3, 4 \tag{4.4}$$

where *d* is 2-dimensional Euclidean distance from event coordinates
r is the radius of influence of the event (degrades over time)
A_k is the maximum penalty value for a k-type event
k = 2 for U-Boats attacked by aircraft
k = 3 for U-Boats killed by aircraft
k = 4 for aircraft sighted by U-Boats

The event penalties (4.4) are constructed to provide an exponentially decreasing penalty extending out from the event coordinates to a certain radius. The initial radius is user-selected and gradually decreases in length over time. This allows the agents to

discount old information, placing greater emphasis on new information. The penalty function provides a penalty that halves (half-life distance) every $\frac{r}{2}$ NM from coordinates of the event.

The penalty for moving to (i, j) is a weighted sum of the component penalties, $P_{i,j}^{<k>}$. The U-Boat agent, then, moves to the coordinates, (i, j), that fulfill

$$\min\left\{\sum_{k=1}^{4} w_k P_{i,j}^{<k>}\right\} \tag{4.5}$$

for integer-valued *i, j* such that $\sqrt{(x_{current} - i)^2 + (y_{current} - j)^2} \leq STEPSIZE$, and w_k is a relative weight given the type-*k* penalty.

In the validation scenarios examined, $w_k = 0$ for k = 2, 3, 4. As a result, U-Boat agents ignore information about contact with aircraft agents and consider only the distances between potential move coordinates and the goal coordinates. Equation (4.5), therefore, reduces to a greedy algorithm for minimizing distance to the agent's goal coordinates. When following this path selection algorithm, the U-Boat chooses an E-W direction of travel. The result, therefore, are U-Boat agents moving as indicated in [McCue, 1990; Waddington, 1973].

The last component of move determination is determining new goal coordinates if $(i, j) = (x_{goal}, y_{goal})$. If the U-Boat has reached its home port, then the new goal coordinates are set to its operational coordinates, and the U-Boat schedules its departure from port according to the in-port maintenance assumptions modeled. If the U-Boat has

reached its operational coordinates, then the agent sets its goal coordinates to its home port and schedules its next update according to supplies remaining and possible resupply.

4.2.4.1.2 Aircraft Behavior

The aircraft search is accomplished via flying to a series of predefined waypoints, in a particular search zone, utilizing a particular search pattern. Each waypoint constitutes goal coordinates the aircraft moves toward sequentially. Aircraft agent behavior consists of a series of decisions that either changes the goal coordinates based on the agent state or allows the goal coordinates to remain the same. The criteria for adjusting the goal coordinates are determined through a hierarchical decision process based on an agent's current state. An aircraft agent makes behavioral decisions according to the hierarchical priorities listed below:

Aircraft Agent Decision Hierarchy

1. Attack U-Boat
2. Search for U-Boat
3. Fuel determination
4. Move

Attack U-Boat. The foremost priority for an aircraft agent is to attack U-Boat agents detected during its search of the Bay of Biscay. If the aircraft agent is within range of a detected U-Boat, signified by collocation of the aircraft and U-Boat agents at a location in the Bay of Biscay, it makes an attack. Attacks varied in effectiveness over the range of the operations, and the particular effectiveness numbers used for model validation are found in Section 4.3.2. If, however, the aircraft is not within attack range

of the U-Boat, the action falls to the fourth decision level (Move). If the aircraft is unaware of any U-Boat location, the decision falls to the second tier of the hierarchy. Following an attack, the aircraft sets its goal coordinates to those of the Allied base, and on each subsequent agent update, enters the decision hierarchy at the fourth level (Move). The aircraft agent is precluded from any action other than a move toward the home coordinates.

Search for U-Boat. If the aircraft agent has not previously discovered a U-Boat, it tries to detect any U-Boats within its combined sensor range. If a U-Boat is detected, the aircraft sets its goal coordinates to those of the discovered U-Boat and proceeds to the fourth tier of the decision hierarchy (Move). Otherwise, the aircraft moves to the third tier.

Fuel determination. If the aircraft has not previously detected a U-Boat and reaches 30% of its original fuel load, it sets its goal coordinates for the home base. At this level of the hierarchy, the aircraft continues to search for U-Boats during subsequent agent updates.

Move. The fourth level of the decision hierarchy determines the coordinates the aircraft agent moves to during the current agent update. The move coordinates are selected via a penalty function evaluation. The aircraft penalty function is a simple 2-dimensional Euclidean distance between the possible move nodes and the aircraft goal coordinates. The aircraft moves to the integer coordinates (i, j) with the penalty value $P_{i,j}$ satisfying (4.6):

$$P_{i,j} = \min\left\{\sqrt{(x_{goal} - i)^2 + (y_{goal} - j)^2}\right\} \tag{4.6}$$

for all integer-valued (i, j) such that $\sqrt{(x_{current} - i)^2 + (y_{current} - j)^2} \leq STEPSIZE$.

The last component of move determination is determining new goal coordinates if $(i, j) = (x_{goal}, y_{goal})$. If the aircraft has reached a waypoint, then the new goal coordinates are set to the next waypoint. If the aircraft has reached the home base, then the aircraft schedules its next search mission and sets its goal coordinates to the first waypoint for its specific search zone and assigned pattern.

4.2.4.2 Aircraft Search

Aircraft agent search was concentrated in a search zone covering the heart of the Bay of Biscay measuring 200 x 350 NM^2 (see Figure 4.4).

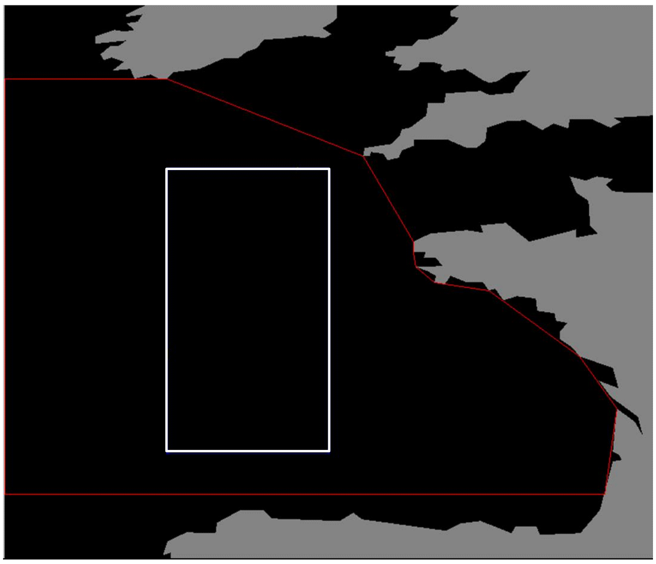

Figure 4.4 Search Zone in the Bay of Biscay

The search zone, in turn, was divided into non-overlapping search grids measuring 50 x 50 NM^2 (see Figure 4.5). Aircraft in the simulation were assigned to a specific grid within which to search for U-Boat agents.

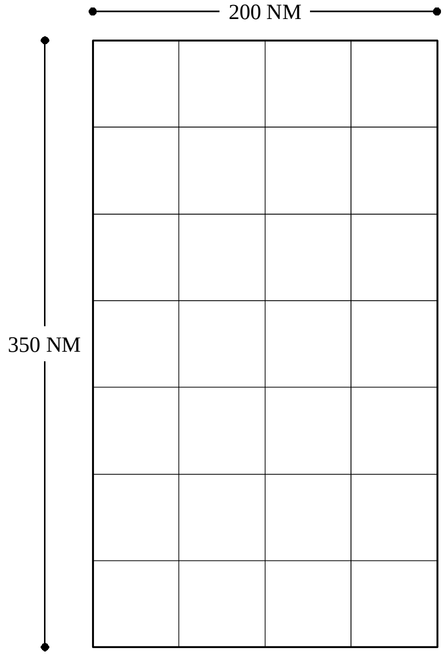

Figure 4.5 Complete Aircraft Search Grid

WW II operations researchers determined that the approach angle optimizing the chance for locating a U-Boat traveling on the surface of the water was a 45° angle [Waddington, 1973]. Since the U-Boats were assumed to move East-West (E-W), searching aircraft would employ SE-NW or NE-SW search lines as much as possible. To this end, a modified barrier search pattern [NCSR, 2000] was simulated for search within each grid (see Figure 4.6). Moreover, the pattern was repeated until the agent either sighted a U-Boat or reached a critical fuel level and returned to base. This search grid size allows multiple passes through the pattern, even for grids remote from the aircraft base.

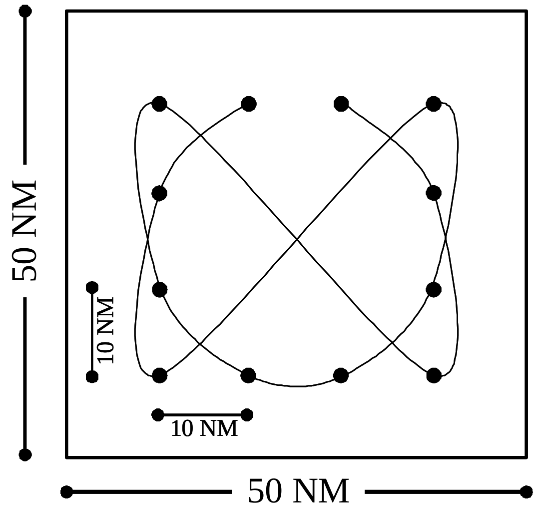

Figure 4.6 Modified Barrier Search Pattern.

Figure 4.7 shows the combinations of these search zone constructs. While the actual size of the operational search grids used by Allied aircraft was not found in the historical record, the agent's searching behavior conforms to historical accounts [Waddington, 1973; McCue, 2002]. Allied pilots were assigned search regions, and pilots repeatedly covered their assigned region until fuel limits forced them to return to their base or until they completed a U-Boat attack. The search zone concept, if not the exact location or size, simulates the historical record as faithfully as the written accounts allow.

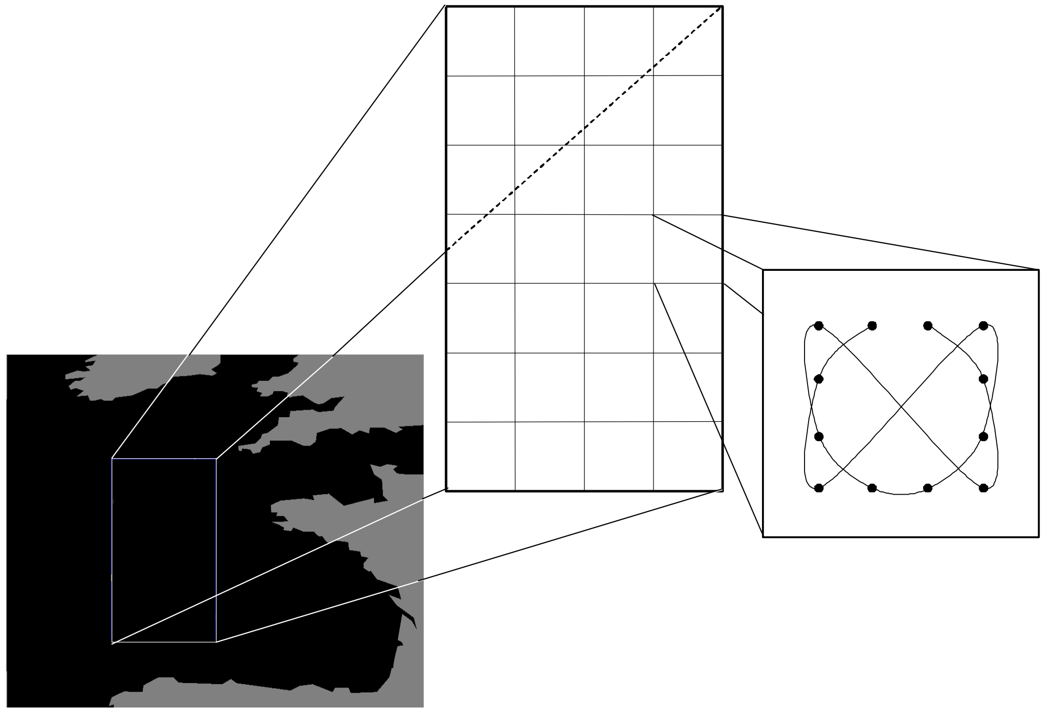

Figure 4.7 Aircraft Agent Search

The aircraft agents were actually capable of flying multiple search patterns within the search zone. In addition to the barrier search pattern used in the model validation effort, each aircraft agent was capable of flying any of five search patterns adapted from the United States National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual [NCSR, 2000].

In search and rescue operations, the NCSR manual acknowledges that choosing an appropriate search pattern for search and rescue operations is highly dependent upon the given scenario. The five search patterns available to each aircraft agent are the parallel, creeping line, square, sector, and barrier search patterns. Each of these is illustrated along with the assumptions under which each is considered the best search option.

When the last point of contact with the search target (datum) is not known with a high degree of certainty and the search area is large, either the parallel (Figure 4.8) or the creeping line (Figure 4.9) search is preferable. The parallel search pattern is most desirable when the target is equally likely to occupy any part of the search area.

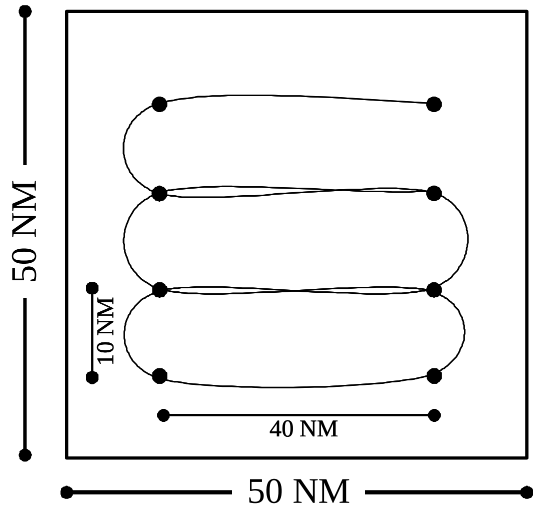

Figure 4.8 Parallel Search Pattern

The creeping line pattern, on the other hand, is typically employed when the target is more likely to be in one end of the search area than in the other. For example, the presence of a current may indicate an increased likelihood of finding the search target toward the down-current portion of the search area. As implemented in (modified for) the Bay of Biscay agent-based simulation, there is no suggestion that the target is located toward one end of the search zone or the other. Therefore, the creeping line pattern resembles the parallel search pattern except the search direction is rotated 90°.

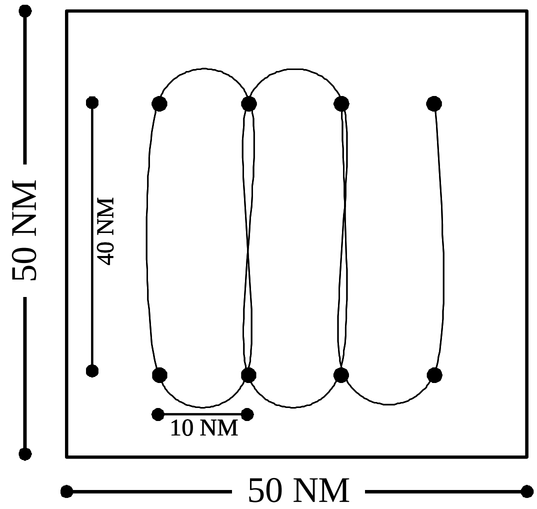

Figure 4.9 Creeping Line Search Pattern

When the point of last contact is well known or established within close limits (i.e. suggesting a relatively small target search area), the square (Figure 4.10) or the sector (Figure 4.11) search patterns are preferable. The square pattern is used when uniform coverage of the search area is desired. The sector search, on the other hand, is used in scenarios where the target is difficult to detect, and the pattern provides for repeated, overlapping coverage of the datum.

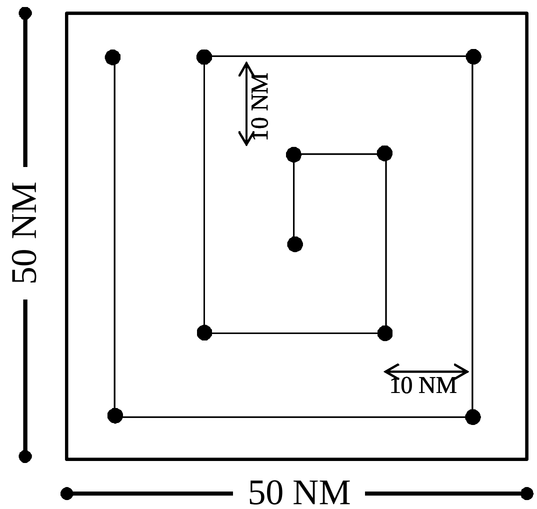

Figure 4.10 Square Search Pattern

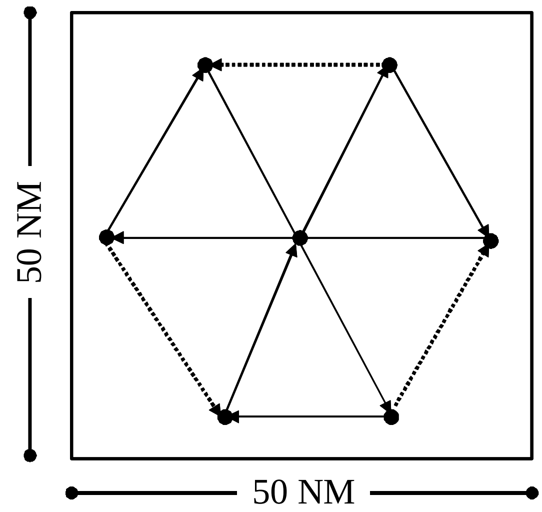

Figure 4.11 Sector Search Pattern

Finally, when the target is fast-moving or when a strong current is present in the search area, the barrier patrol search pattern (Figure 4.6) is the preferred search pattern. The pattern provides concentrated search around the perimeter of the search zone with repeated revisiting of the datum.

In addition to the capability to fly the above five search patterns in the non-overlapping search zones, each aircraft agent was able to perform the search using an overlapping search zone grid. In the overlapping search, the search zones measured 100 x 100 NM^2 and overlapped each of the adjacent zones by 50 NM. The search region was the same as depicted in Figure 4.4 and Figure 4.7 resulting in 18 search zones contained within the region. Again, each of the five search patterns was available within each of the overlapping search zones.

Inherent in search and rescue operations is the assumption of a cooperative target, that is, the target of the search is either actively working to aid detection during the

search or, at the very least, not actively trying to avoid detection. The Bay of Biscay search scenario involves uncooperative search targets (i.e. the U-Boats are actively acting to avoid detection). Results comparing the search effectiveness of all five search patterns in both the non-overlapping search zones and the overlapping search zone are reported in [Champagne, *et al*, 2003a; Champagne, *et al*, 2003b; Carl, 2003].

For the purposes of model validation versus the historical record, the modified barrier search pattern was selected as the pattern most likely to conform to the historical accounts, and thus it is the sole pattern used.

4.2.4.3 Agent Strategy and Adaptation

The strategies of both the aircraft and U-Boat agents are based on the possibility of interaction between the opposing agents. The aircraft agents want to maximize the chance of finding (interacting with) a U-Boat. The U-Boat agents want to minimize their chances of coming into contact with the aircraft. Specific strategic behaviors for each of the agent types are illustrated below.

Given that the search pattern and search zone for the aircraft agent are set for the historical validation, the primary strategic consideration remaining is the timing of the search. WWII planners had to take into account the possible reactive strategy of the enemy. For example, if the aircraft concentrated their search exclusively during the daylight hours, the U-Boats could surface exclusively during the nighttime hours to avoid the searchers. Conversely, if searches were conducted exclusively during nighttime hours, the U-Boats could counter with a daytime-only surfacing policy that would guarantee no contact between the opposing forces. Therefore, the aircraft were forced to

conduct searches throughout all parts of the day [McCue, 1990]. For the purposes of simulating history, the aircraft were distributed (scheduled) for takeoff randomly throughout each 24 hour day.

U-Boat agent strategy concerned two principal factors: time spent on the surface of the bay and time of day to surface. Traveling on the surface more than was necessary to charge their batteries dramatically reduced the time needed to cross the bay. However, traveling along the surface made the U-Boats vulnerable to detection and attack from the Allied aircraft. The U-Boat fleet experimented with surfacing only at night to reduce the threat of attack versus surfacing when needed in order to move across the bay and into the operational zone more quickly. This, however, had an operational impact in that waiting for a particular time of day (i.e. nighttime) to surface could delay the crossing, thereby reducing the time the U-Boat could spend in the North Atlantic searching for Allied transport ships.

The U-Boat fleet used both extremes of this surfacing policy in crossing the Bay of Biscay at various times during the conflict. Under a policy of maximum submergence, the U-Boats would surface only enough to recharge their batteries before submerging again to continue their crossing. At other times, the U-Boats attempted to “race” across the bay to the North Atlantic, submerging only when coming into contact with an Allied aircraft.

The second U-Boat policy decision, daytime/nighttime surfacing, directly plays against the aircraft search strategy. If the U-Boats concentrated their surfacing during one part of the day, then the aircraft could synchronize their search to coincide with the

surfacing. The historical record shows that the U-Boat fleet policy used nighttime-only and surface-as-required surfacing policies at different times of the conflict trying simultaneously to minimize U-Boat vulnerability while maximizing U-Boat concentration in the North Atlantic [McCue, 1990].

The two scenarios chosen for validating the model performance versus the historical outcomes (section 4.3.2) were chosen, in part, because the U-Boat fleet policies during these times were at extremes with respect to these two policy parameters. The U-Boat agents follow the fleet policy known to be in effect during the time simulated. For example, Scenario 1 (October 1942 – March 1943) simulates a fleet policy of maximum submergence and nighttime surfacing. During a period of "maximum submergence," U-Boats travel on the surface of the bay only long enough to charge their batteries and only by night. Similarly, under the nighttime-only surfacing policy, the U-Boat agents only surface during the time between the end of nautical dusk and the beginning of nautical dawn. Scenario 2 (April 1943 – September 1943) employs surface-only movement during the day and mandated submergence during the nighttime. Under this policy configuration, the agents only submerge during the daytime when they come into contact with an Allied aircraft agent. Once submerged, they travel the full extent allowed by their batteries before resuming surface travel. The U-Boats in Scenario 2 only surface during the hours between nautical dawn and nautical dusk.

Hill, *et al*, (2003a), demonstrated the interplay between aircraft agent search and U-Boat agent surfacing strategies within a game theory construct by allowing the agents to adapt their strategies based on their collective experiences. The experiment allowed

for single-sided adaptation as well as simultaneous adaptation (see section 4.2.5). In each case, the results were indicative of those expected under game theory.

4.2.4.4 Other Agent-Based Issues

Simulations often rely on common random numbers as a variance reduction technique. Depending on the agent implementation, this may or may not be possible. For instance, in a multi-threaded design, it is highly unlikely that agent threads act in precisely the same order throughout the course of all replications. Moreover, depending on the operating system, the thread handling is often an uncontrolled stochastic process.

Attempts at controlling agent processing, however, tends to reduce the autonomy associated with the actions of each individual agent, tending to move the simulation entities away from the definition of agent. Therefore, the analyst is left with little option outside of increasing the number of replications in order to reduce variance within the simulation.

4.2.4.5 Model Verification

As with any software project of significant complexity, an extensive number of verification techniques were used to ensure the executable model represented faithfully the conceptual model. Verification methods were used from all three categories of the verification taxonomy presented in the previous chapter of this document. The most significant of these are presented below.

4.2.4.5.1 Good Software Engineering Practices

Among the many good software engineering practices used to verify the translation of the model to an executable form, OO design and use of a development environment were the most significant.

4.2.4.5.1.1 Object-Oriented (Modular) Design

The Bay of Biscay agent-based simulation was developed in JAVA, a pure OO language. Because of the JAVA language requirements, the variables are strongly typed and the resulting code is necessarily completely object oriented. Designing for an agent-based simulation, however, required additional modularity above that called for by the development language. Specifically, each agent is designed as a separate object.

Individual agent behaviors were developed modularly. Developing the methods within the construct of the agent shell provided the necessary framework for mixed (bottom-up and top-down) testing mentioned in [Sargent, 1996]. The agent object provided a natural harness for verification testing of the various methods affecting the agent's behavior as they were developed.

The simulation was designed to take maximum advantage of the OO property of inheritance. Inheritance allows similar objects to be derived from a base object. The attributes and methods similar to all derived objects are found in the base object, while those methods and attributes that distinguish between different derived objects are extended from the base object and found only in the code for the derived object. This has the effect of reducing the verification effort necessary.

For example, though the U-Boat and aircraft agents were ultimately distinct, each had common attributes (e.g. positional coordinates, goal coordinates, etc.) and methods (e.g. thread start, thread stop, reset after each replication, animation translations, etc.). Therefore, a base agent was constructed having the common attributes and methods. Within the base object, the common modules could be verified in a single effort rather than twice (as would have been the case without the base class and inheritance).

4.2.4.5.1.2 Use of Development Environment

The Bay of Biscay agent-based simulation was coded within the Sun One ®, Community Edition JAVA development environment. This provided several advantages over coding in a text editor. The primary advantage is the syntactical checking that occurred as the code was entered. Individual statements were interpreted for correct syntax as they were typed, thereby providing immediate indicators when the syntax was incorrect. Other tools included automated indentation of nested statements and highlighting of the alternate parenthesis or bracket from the other in the pair. Together these tools minimized the time necessary in debugging the syntax and allowed more time to be spent verifying the logic of the code.

Additionally, the JAVA language provides for the generation of automated hypertext documentation through special internal comment placement and code markers. The Bay of Biscay agent-based simulation was coded with extensive use of these comments, which facilitated the static verification (and code alteration when necessary).

4.2.4.5.2 Static Verification

Static verification is done prior to code execution and was achieved through two primary tools: through the use of the JAVA compiler and with formal and informal walkthroughs. Each of these is detailed individually.

4.2.4.5.2.1 Static compilation

Compilers translate text-based computer code into machine executable code. The JAVA compiler also provides additional functionality in static verification. First, the compiler identifies variables that are used prior to initialization, preventing one possible source of numerical error. Second, the code is examined for logical completeness, and the compiler identifies logical branching that is incomplete. Third, the compiler identifies sections of code that are inaccessible under any circumstances. These functions help minimize the most common logical errors in coding the simulation.

4.2.4.5.2.2 Code/logical walkthrough

Each module was designed using logic flow diagrams and pseudo-code prior to coding in the development environment. Depending on the complexity of the method being developed, these diagrams and pseudo-code modules were subjected to either informal or formal walkthroughs. Informal walkthroughs were of the desktop variety, while formal walkthroughs consisted of up to three individuals familiar with the project in addition to the developer. Formal walkthroughs were held as often as weekly during the most intensive four months of the simulation development. Following a successful walkthrough, the pseudo-code was translated into JAVA code, compiled, and dynamically tested.

4.2.4.5.3 Dynamic Verification

Dynamic verification is performed while the model is executing. The following sections highlight the most important tools used to verify the Bay of Biscay agent-based simulation model.

4.2.4.5.3.1 Animation

Animation during program execution provided verification for nearly all of the agent behavior found in the simulation. Through the visualization of the agents, logical errors were detected for subsequent correction in a number of situations including incomplete reset between replications, inappropriate submergence behavior, stationary agents due to incomplete movement logic or unforeseen events, and numerous other faults that typically occurred at decision points for the agents.

Even though the animation was an important first indicator of logical errors, an animation tool provides only a coarse level of verification. Several classes of problems are not identifiable through animation. This is true for a number of reasons including: the problem occurs when the agent is not visible; the behavior of the agent seems reasonable, but it is not the behavior that was intended under a specific circumstance; or the troublesome event occurs too infrequently to be spotted during small verification runs. Other techniques were used to get finer verification resolution.

4.2.4.5.3.2 Trace output, model instrumentation, and debugging.

The most extensive dynamic verification tool used was model instrumentation and output tracing. As each new module was incorporated into the simulation, lines of code were added to output both the environmental and individual agent states at particular

events (e.g. reaching port or base, sighting or attacking a U-Boat, or change in submergence status). The output state values and corresponding agent behaviors could then be scrutinized for consistency with the conceptual model.

The development environment provided a debug mode, which provided a similar framework for verification. During simulation execution, attribute watches could be set along with break points enabling a more flexible method of monitoring agent and environmental states. Unlike the model instrumentation, these could be changed during execution and linked to a specific agent of interest.

4.2.5 Agent Adaptation

The Bay of Biscay scenario contains several interesting conflicting strategies for each side in the operation. One of the more important strategies involved day versus night considerations. The Allied aircraft search effort desired maximum contact and kills of U-Boats. The U-Boat fleet's surfacing policy sought to minimize the vulnerability of the fleet.

Consider for example that aircraft attacks were dramatically more successful during the daytime hours. Allied forces thus would prefer predominantly daytime attacks. However, concentrating all aircraft sorties during the daytime hours would allow the U-Boats to surface exclusively during the nighttime hours effectively negating the entire Allied search effort. Conversely, concentrating search activity during nighttime hours gives U-Boats a counter of surfacing during the daytime hours, again negating the Allied strategy. Therefore, the Allied search required both daytime and nighttime effort to prevent the U-Boat surfacing policy from adapting to the Allied search strategy.

The agents in the Bay of Biscay agent-based simulation were provided an adaptive capability. The adaptation was designed around the day versus night strategy. Aircraft agents used their collective experiences to apportion search effort between daytime and nighttime searches in an attempt to increase the level of contact (and kills) with the U-Boat fleet. U-Boat agents used their collective experiences to adjust their surfacing policy to reduce the level of contact with Allied aircraft, thereby countering the perceived Allied strategy.

The Bay of Biscay agent-based simulation could be set to allow: 1) no agent adaptation; 2) Aircraft-only adaptation; 3) U-Boat only adaptation; or 4) two-sided adaptation (co-evolution). Historical validation efforts were made with no agent adaptation. The effect of adaptive strategies (configurations 2, 3, and 4, above) was explored in the context of a game theory framework, and the results are reported in (Price, 2003; Hill, *et al*, 2003a).

4.2.5.1 Aircraft Adaptation

Aircraft adaptive strategy involved the apportionment of search effort between daytime and nighttime search. The aircraft agent has complementary probabilities of scheduling daytime (P_{day}) or nighttime ($P_{night} = 1 - P_{day}$) missions. Each aircraft schedules its "next" mission according to a random draw against P_{day}. Given a uniform random draw, U, such that $U \leq P_{day}$, the aircraft will schedule itself for a search during the next daytime period; otherwise, it will schedule itself for a nighttime search during the next period.

In using the P_{day} versus P_{night} construct, the number of sorties scheduled for daytime searches is a random variable. The expected number of the sorties scheduled for daytime search is given by

$$E[S_{day}] = P_{day} \cdot S_{total} \tag{4.7}$$

where S_{day} is the fraction of scheduled sorties performing daytime searches and
S_{total} is the total number of sorties scheduled.

Similarly, the expected number of nighttime search sorties scheduled is given by:

$$E[S_{night}] = (1 - P_{day}) \cdot S_{total} = P_{night} \cdot S_{total} \tag{4.8}$$

where S_{night} is the fraction of scheduled sorties performing nighttime searches and
S_{total} is the total number of sorties scheduled.

For daytime searches, the aircraft agent scheduled its takeoff time uniformly over the period from three hours prior to sunrise to seven hours prior to sunset. The time window prior to sunrise provides sufficient time for ingress to the search zone prior to the start of its search. Similarly, the seven hour limit with respect to sunset provides enough time to search within the assigned search zone prior to night fall. For nighttime scheduling, an aircraft agent selects a takeoff time uniformly over the time period from three hours prior to sunset until seven hours prior to the following sunrise. Again, the three hours prior to sunset allow sufficient time for ingress to the search zone to allow searching to begin as soon as the sun sets. The seven hours limiting takeoffs prior to sunrise ensures sufficient mission duration to provide effective search within the search zone for the missions scheduled for the later portion of the nighttime. Figure 4.12 is a generic representation of the scheduling process for both day and night search missions.

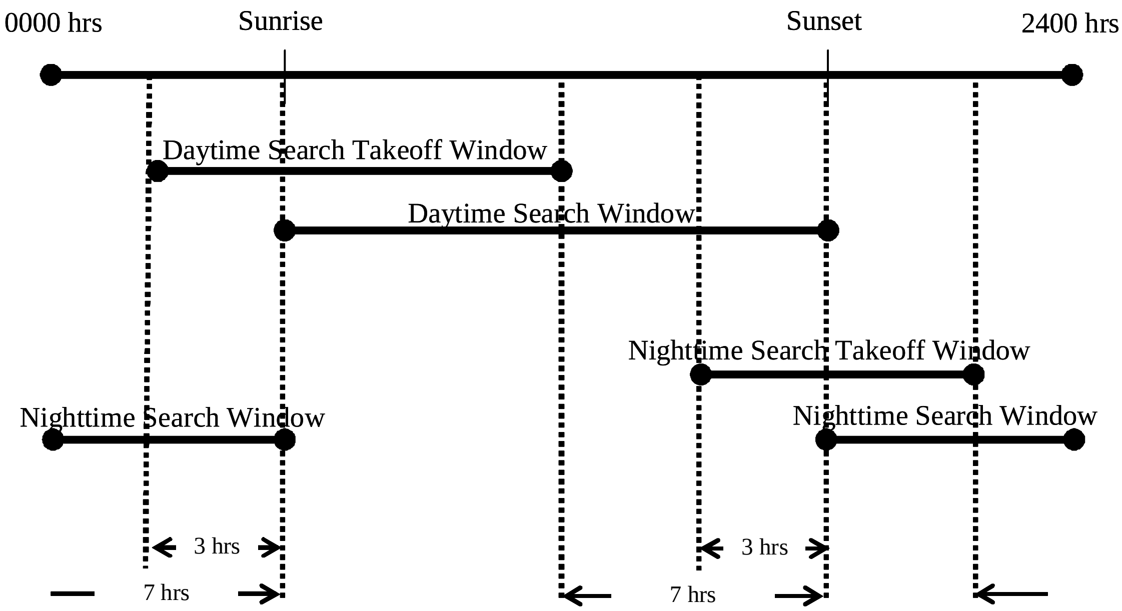

Figure 4.12 Generic Aircraft Agent Scheduling Process for Day versus Night Missions

The aircraft fleet collects information on U-Boat sightings based on daytime versus nighttime contact. The collected information allows aircraft agents to modify their P_{day} and P_{night} to improve their perceived chances of making contact with the evasive U-Boat fleet.

The adaptation algorithm adjusts the value, P_{day}, at equal time increments and is a two step process. Step 1 computes the fraction of U-Boat sightings during the ith time period occurring during the daytime:

$$f_i = \frac{S_{day_i}}{S_{day_i} + S_{night_i}} \tag{4.9}$$

where S_{day_i} is the number of daytime sightings during the ith time period, and

S_{night_i} is the number of nighttime sightings during the ith time period.

Step 2 computes P_{day} for the (i + 1)th time period as a weighted average of P_{day_i} and f_i.

$$P_{day_{i+1}} = \frac{(i) \cdot P_{day_i} + f_i}{i+1} \tag{4.10}$$

$$P_{night_{i+1}} = 1 - P_{day_{i+1}} \tag{4.11}$$

The algorithm contains two special cases in the event that $S_{day_i} + S_{night_i} = 0$. If there were no sightings during the ith time period and $P_{day_i} = 1.0$ or $P_{day_i} = 0.0$, then $P_{day_{i+1}} = 0.5$, otherwise $f_i = 0.5$ in an attempt to move $P_{day_{i+1}}$ toward a value likely to provide some contact with the U-Boats . In either case, P_{day_0} is an initial setting defined by the model user.

The advantage of the weighted average approach to search strategy adaptation is two-fold. First, in the initial stages of the conflict, the aircraft strategy cannot move more than half the distance to the observed fraction of sightings, thereby preventing overcompensation for sightings that, through random occurrence, do not accurately reflect the U-Boat surfacing strategy. Second, as the aircraft strategy matures, the current strategy becomes more important, thereby stabilizing the adaptation process, leaving just fine tuning of the probability values.

4.2.5.2 U-Boat Adaptation

U-Boat adaptive strategy involved apportioning the fleets surfacing between daytime and nighttime when a U-Boat is within the Bay of Biscay and vulnerable to attack from Allied aircraft agents. The U-Boat strategy was expressed through a

complementary pair of probabilities P_{day} and P_{night}. To exercise the strategy, a U-Boat needing to surface makes a uniform random draw, U, against P_{day}. If it was daytime and $U \le P_{day}$, then the U-Boat surfaces; otherwise, it stays submerged and surfaces as soon as sunset had occurred. If it was nighttime and $U > P_{day}$, then the U-Boat surfaces; otherwise, it stays submerged and surfaces as soon as sunrise had occurred. The check is made each time the U-Boat attempts to surface after traveling the extent of its battery reserves underwater.

The U-Boat adaptation algorithm differs from the aircraft. The U-Boat strategy was built around decreasing the number of contacts between the opposing sides. The U-Boats also track aircraft sightings prior to discovery of the U-Boat by the aircraft. Finally, the U-Boats consider the fraction of kills made during the daytime and nighttime in addition to the fraction of daytime versus nighttime U-Boat sightings by aircraft.

The U-Boat strategy adaptation algorithm adjusts P_{day} in equal time increments. The U-Boat algorithm is a three step process. Step 1 computes the fractions of the three contact types, index *j*, during the i^{th} time period occurring during the daytime:

$$f_i^{<j>} = \frac{S_{day\ i}^{<j>}}{S_{day\ i}^{<j>} + S_{night\,i}^{<j>}} \tag{4.12}$$

where $S_{day\ i}^{<j>}$ is the number of daytime j-type contacts during the i^{th} time period,

$S_{night\,i}^{<j>}$ is the number of nighttime j-type contacts during the i^{th} time period, and

j = 1, 2, 3 represents U-Boats sighted by aircraft, U-Boats killed, and aircraft sighted by U-Boats, respectively.

Step 2 computes a weighted sum of $f^{<j>}$:

$$f_i = \sum_{j=1}^{3} w_j \cdot f_i^{<j>} \tag{4.13}$$

where w_j is the weight given the j^{th} contact type and $\sum_{j=1}^{3} w_j = 1$.

Step 3 computes P_{day} for the $(i + 1)^{th}$ time period as a weighted average of P_{day_i} and f_i.

$$P_{day_{i+1}} = 1 - \frac{(i) \cdot (1 - P_{day_i}) + f_i}{i + 1} \tag{4.14}$$

$$P_{night_{i+1}} = 1 - P_{day_{i+1}} \tag{4.15}$$

Comparing the two adaptation algorithms, naturally, the aircraft adaptation algorithm tends to move the aircraft agents toward more contact with the opposition, while the U-Boat algorithm tends to favor fewer contacts with the Allied aircraft agents.

4.2.6 Simulation Output Format

The Bay of Biscay agent-based simulation tracks multiple measures of effectiveness throughout the duration of the runs. The data is organized by month and by simulation replication (iteration), so for each simulation run, each MOE (i.e. aircraft flying hours, U-Boats sighted, and U-Boats killed) is output as a matrix, **X**, such that for each MOE, $x_{i,j}$ is the value of the MOE for the i^{th} replication during the j^{th} month. Two scenarios were run (see section 4.3.2). Each scenario simulated 6 months (j = 1, 2, …, 6) and was replicated 20 times (i = 1, 2, …, 20).

From this matrix, multiple significant measures can be derived for useful analysis. The most obvious of these are presented in the remainder of this section and are presented assuming 20 replications of a 6-month simulation experiment.

4.2.6.1 Iteration Total

The total value of the MOE for the i^th replication is:

$$x_i = \sum_{j=1}^{6} x_{i,j} \tag{4.16}$$

4.2.6.2 Mean Total Value

The mean total MOE value over all replications is:

$$\bar{x} = \frac{1}{20} \sum_{i=1}^{20} x_i \tag{4.17}$$

4.2.6.3 Iteration Mean Monthly Value

The mean monthly value of the MOE for the i^th replication is:

$$\overline{x_i} = \frac{1}{6} \sum_{j=1}^{6} x_{i,j} \tag{4.18}$$

4.2.6.4 Overall Mean Monthly Value

The overall mean of monthly value of the MOE is:

$$\bar{\bar{x}} = \frac{1}{20} \sum_{i=1}^{20} \overline{x_i} \tag{4.19}$$

4.3 Analysis Objectives

The first step in the modeling process was to determine the analysis objectives for the simulation development. The primary objective was to demonstrate that agent-based combat simulation could be sufficiently advanced to mission-level combat modeling. In making this determination, the model would be subjected to validation techniques comparing the simulation output to a known historical scenario. Analysis techniques were developed to compare the historical and model results.

The determination of whether or not a model is validated is necessarily a subjective function of intended model use. The required accuracy for model output is also subjectively determined by the level of risk inherent in accepting output from a model that may be incorrect. The validation criteria used to demonstrate sufficiency for the Bay of Biscay agent-based simulation provides a statistical argument against invalidating the model with respect to the historical scenarios. That is, can an agent-based model of the offensive search operations in the Bay of Biscay come sufficiently close to the historical outcomes to prevent statistical rejection at a reasonable confidence level?

In addition to model validation, the Bay of Biscay agent-based simulation was to be used in two other demonstrations of capabilities in other areas of research. First, the simulation was used to determine the applicability of agent-based combat simulations to provide insight into offensive search techniques, demonstrating the ability to differentiate between various search strategies. Second, the model was used in an analysis of

agent-based results with respect to game theory principles, specifically demonstrating the effects of strategy adaptation on the part of both agent types on the scenario MOEs.

The complement of model specifications were derived based on the needs of each of the three analysis objectives specified above.

4.3.1 MOEs

Output from the Bay of Biscay agent-based simulation are compared to the two primary measures of effectiveness (MOEs) from the real-world data, number of U-Boats sighted and number of U-Boats killed (sunk).

While the validation using simulation MOEs gives confidence as to the validity of the model, there are other agent-based characteristics that should be tested, specifically any emergent behavior from the model. As an example, a secondary measure, the distribution of U-Boat arrivals into the Bay of Biscay, is addressed as a validation measure of emergent agent behavior. Operational analysts noted that the U-Boats entered the Bay of Biscay according to a Poisson distribution [McCue, 1990; Waddington, 1973]. The simulation model made no effort to force the U-Boat agents into specific behavior to conform to a Poisson arrival distribution, or in fact to any particular distribution. Therefore, the arrival times in the bay are an emergent phenomenon.

4.3.2 Validation Scenarios

Two scenarios were chosen for validating the simulation. The first was the six month period from October 1942 – March 1943 (henceforth, Scenario 1), and the second was a six month period from April 1943 – September 1943 (Scenario 2). These scenarios

were chosen because the technologies used by both Allied aircraft and German U-Boats remained relatively constant over the time period, but were different between scenarios. Moreover, the German U-Boat command's submergence policy used by the U-Boat captains within each scenario was stable meaning the fleet behaved consistently throughout each period.

Scenario 1 (October 1942 – March 1943) simulates a U-Boat fleet policy of maximum submergence and nighttime surfacing [McCue, 1990]. Under this policy, U-Boat agents will travel on the surface of the bay only long enough to charge their batteries and only by night. The U-Boat agents in Scenario 1 will only surface during the time between the end of nautical dusk and the beginning of nautical dawn.

Scenario 2 (April 1943 – September 1943) employs a U-Boat fleet policy of surface-only movement during the day and mandated submergence during the nighttime [McCue, 1990]. Under this policy configuration, the U-Boat agents will only submerge during the daytime when they come into contact with an Allied aircraft agent. Once submerged, they will travel the full extent allowed by their batteries before resuming surface travel. The U-Boat agents in Scenario 2 will only surface during the hours between nautical dawn and nautical dusk.

The U-Boat fleet initially consists of 70 agents distributed randomly and uniformly throughout the Bay of Biscay, half of the fleet moves toward the North Atlantic, and half moves toward their home port. There are five home ports located on the coast of France, and the agents are evenly assigned among them. This initial U-Boat agent configuration was not representative of usual operations.

A simulation warm-up period of 12 months is used to position the fleet, through normal movement through the bay and time spent in operational zones and ports, in a more natural configuration as might have been the real-world case. During the warm-up period, the aircraft do not hunt the U-Boats. U-Boat fleet reinforcements begin arriving in the North Atlantic from Germany according to their historical numbers [McCue, 1990] in month 12 of the warm up period and continue throughout the remainder of the simulation (Table 4.1). The U-Boat reinforcements are divided evenly between four of the five French ports.

Table 4.1 U-Boat Reinforcements for Validation Scenarios [McCue, 1990]

Scenario 1			Scenario 2	
Enter Simulation	Number of U-Boats		Enter Simulation	Number of U-Boats
Sept 1942	32		Mar 1943	25
Oct 1942	32		Apr 1943	13
Nov 1942	27		May 1943	22
Dec 1942	11		Jun 1943	16
Jan 1943	14		Jul 1943	7
Feb 1943	14		Aug 1943	3

The literature does not report the number of aircraft conducting offensive search operation during each scenario. However, the number of flying hours during each scenario is reported. Therefore, the number of aircraft agents within each scenario was set to agree with the historic sortie hour levels recorded during the time periods modeled. The modeled aircraft fleet consists of 19 agents in Scenario 1 and 31 agents in Scenario 2, operating from a single airbase in Great Britain. The number of aircraft agents remains constant throughout each scenario simulated.

Aircraft offensive search is assigned to a fixed area of the bay 200 x 350 NM^2 (E-W x N-S) (see Figure 4.4). The search area is subdivided into 50 x 50 NM^2 non-overlapping grids (see Figure 4.5). Aircraft search each grid according to a modified barrier search pattern constructed from the tactics discussed in [Waddington, 1973] (see Figure 4.6). In addition, the aircraft search for U-Boats during ingress to and egress from their assigned search area.

Aircraft attacks varied in effectiveness in each of the scenarios. The aircraft attack effectiveness (P_k) during Scenario 1 was computed as the ratio of kills to sightings as found in [McCue, 1990], resulting in a $P_k = 0.02$. No data was available to allow distinction between daytime and nighttime effectiveness for Scenario 1. Waddington (1973) presented aircraft attack effectiveness for the time period covered by Scenario 2 and further differentiated between daytime and nighttime effectiveness. The model incorporated the Waddington material as nighttime $P_k = 0.11$ and daytime $P_k = 0.4$.

For validation purposes, each scenario was replicated 20 times, and statistics were kept for the 6-month total and on a per-month basis. The number of replications was selected based on the stability output variance. Prior to production runs, both scenarios were run over varying numbers of replications and resulting variances calculated. Theses results are plotted in Figures 4.13 and 4.14 for Scenario 1 and Scenario 2, respectively. As shown in the figures, the output variance was fairly stable after ten replications yielding twenty replications as a final replication number for the research production runs.

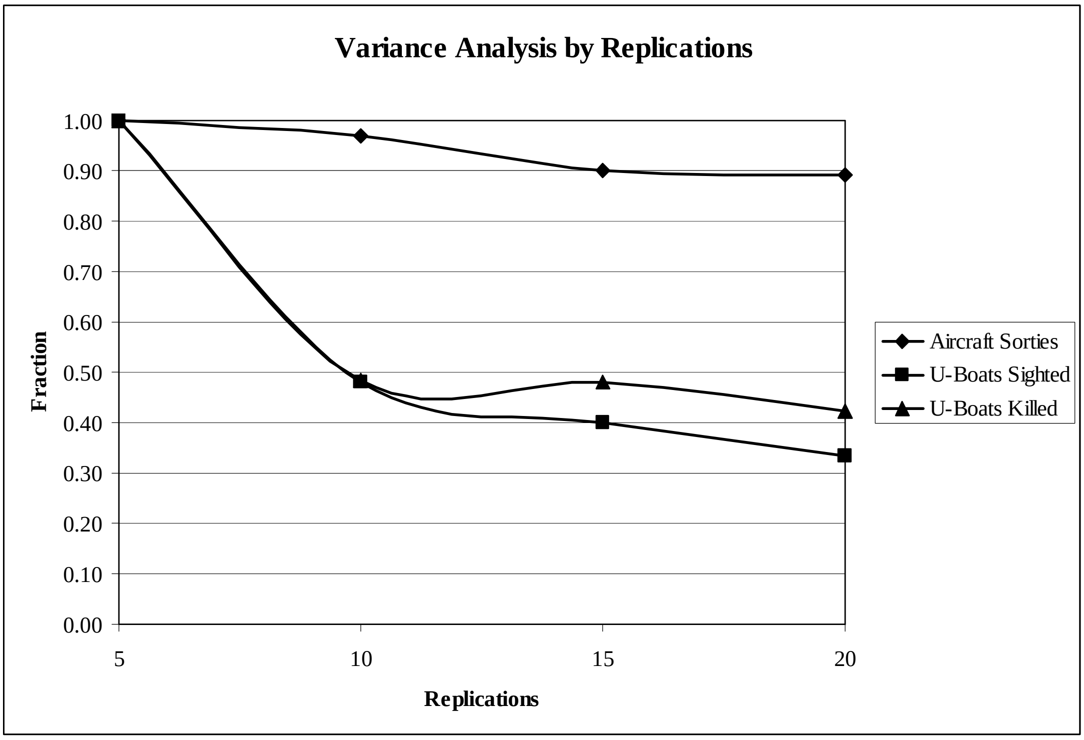

Figure 4.13 Variance Reduction in Pre-Production Model, Scenario 1

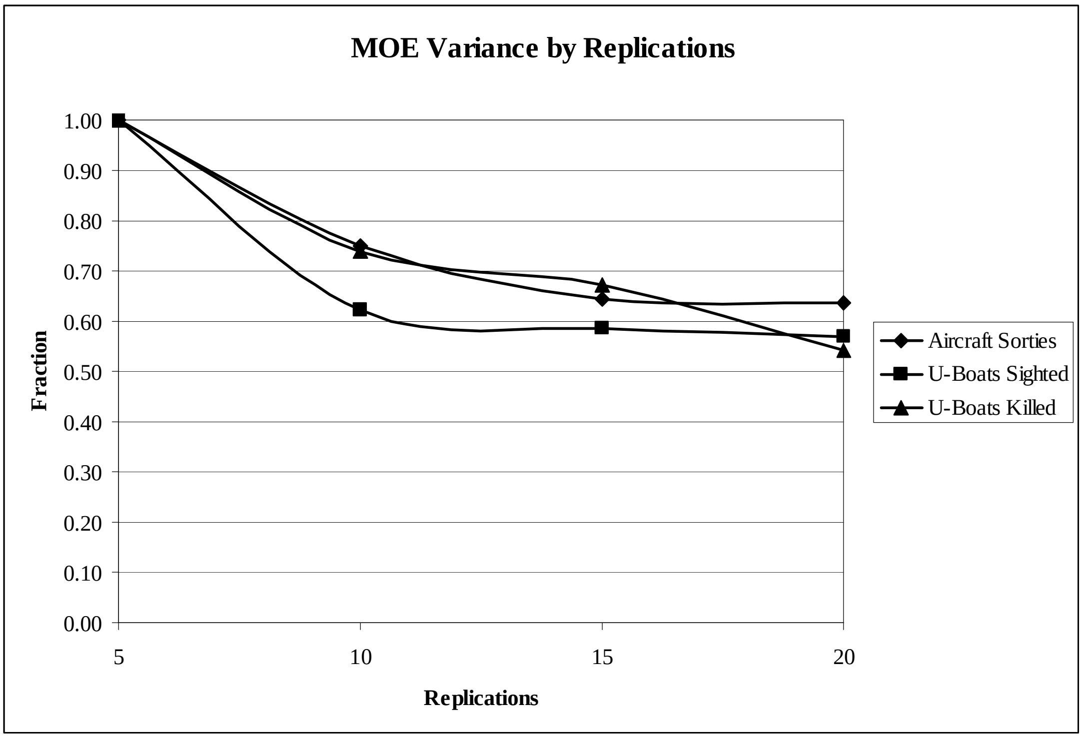

Figure 4.14 Variance Reduction in Pre-Production Model, Scenario 2

4.3.3 Validation Criteria

There are particular considerations when building a model based on an historical operation. The extent to which the model is a valid representation of the real-world system is directly related to the proximity of the simulation output to the real-world MOE values. However, the simulation is an approximation of the real-world system, and is unlikely to match the real-world system exactly. How close, then, is close enough?

The validation literature lacks a definitive answer to the above question. "Close enough" is both simulation and circumstance dependent. The answer depends on a number of factors including risk associated with using an incorrect model and the fidelity of (or confidence in) the inputs that drive the model performance. In this research's case study, success is defined as follows: given a level of effort for offensive search reasonably close to the level of effort expended during the simulated periods of time, the simulation produces results similar to those produced in the real-world scenario. Limitations in the fidelity of the input data, specifically the Allied level of effort (sortie hours) necessitate this broad definition. Section 4.4 builds a case for accepting the Bay of Biscay agent-based simulation as a valid representation of the real-world operations accordingly.

Validating the simulation against the historical record raises another serious issue for combat simulations. Acknowledging that true combat is a stochastic process, a single historic combat result represents a potentially dangerous comparison. If the event is compared to a simulation mean, then the results from the real-world event is implicitly

taken as the mean of all possible real-world outcomes. With only one sample for comparison, there is no way to know the fitness of this assumption.

Though such a comparison is risky, and statistically suspect, the single real-world conflict is the best guess for the mean when there has only been the one conflict. The validation in this effort uses the real-world data in such a manner. That is, the real-world data is assumed to be the true mean of all combat under the same conditions.

4.4 Model Output and Validation

Table 4.2 and Table 4.3 show the real-world MOE values for Scenario 1 and Scenario 2, respectively. MOE values for each month of the operation were taken from [McCue, 1990]. The values in the column under "Sum" represent the totals for each MOE over the entire time period and were computed using (4.16). Likewise, monthly means for each MOE were computed using (4.18) and can be found under the "Mean" heading.

Table 4.2 Historical MOE values for Scenario 1 [McCue, 1990]

MOE	Oct 42	Nov 42	Dec 42	Jan 43	Feb 43	Mar 43	Sum	Mean
Sortie Hours	4,100	4,600	3,400	3,130	4,400	4,600	24,230	4,038.3
Sightings	18	19	14	10	32	42	135	22.5
Kills	1	1	0	0	0	1	3	0.5

Table 4.3 Historical MOE values for Scenario 2 [McCue, 1990]

MOE	Apr 43	May 43	Jun 43	Jul 43	Aug 43	Sep 43	Sum	Mean
Sortie Hours	4,200	5,350	5,900	8,700	7,000	8,000	39,150	6,525.0
Sightings	52	98	60	81	7	21	319	53.2
Kills	1	7	4	13	5	2	32	5.3

4.4.1 Gauging the Allied Level of Effort

The level of effort for each simulated scenario was determined by adjusting the number of aircraft agents acting within the simulation until the total number of sortie hours simulated was in a reasonably close neighborhood to the actual sortie hours flown. Inspection of the monthly sortie hour values in Table 4.2 and Table 4.3 shows that the number of sortie hours stated for each month is an even multiple of 10, and if the records were accurate, the numbers would probably show less consistency. In all likelihood these numbers are rounded or approximated.

It is impossible, therefore, to know the true value of sortie hours flown (though the reported values are still termed “actual” or “real-world”), and this supports why a more exacting standard was not used. Table 4.4 shows the simulated sortie hours for Scenario 1, including the corresponding total level of effort and mean monthly sortie hours. Table 4.5 shows the same data for Scenario 2.

Table 4.4 Simulated Aircraft Sortie Hours for Scenario 1

	Oct 42	Nov 42	Dec 42	Jan 43	Feb 43	Mar 43	Sum	Mean
Iteration 1	3,206	3,487	3,651	3,208	2,702	3,970	20,224	3,371
Iteration 2	4,059	3,742	3,932	3,805	3,001	3,399	21,938	3,656
Iteration 3	4,404	4,146	4,080	3,945	3,692	4,493	24,760	4,127
Iteration 4	4,333	4,137	4,222	4,189	3,532	4,010	24,423	4,071
Iteration 5	3,749	4,043	3,911	3,402	3,687	3,612	22,404	3,734
Iteration 6	3,782	3,816	3,865	3,952	3,208	3,809	22,432	3,739
Iteration 7	4,162	3,969	4,238	4,175	4,006	4,037	24,587	4,098
Iteration 8	4,428	4,182	4,078	4,217	3,812	4,264	24,981	4,164
Iteration 9	4,146	4,202	4,360	4,200	4,001	4,136	25,045	4,174
Iteration 10	4,391	4,180	4,135	4,257	3,964	4,034	24,961	4,160
Iteration 11	3,553	3,388	3,543	2,399	3,198	3,851	19,932	3,322
Iteration 12	3,745	3,747	3,848	3,941	3,182	4,266	22,729	3,788
Iteration 13	3,871	3,041	3,276	3,519	2,667	4,119	20,493	3,416
Iteration 14	3,692	4,194	3,142	3,651	3,538	3,726	21,943	3,657
Iteration 15	3,969	3,673	3,818	3,446	3,568	3,934	22,408	3,735
Iteration 16	4,046	3,955	4,097	3,813	3,287	4,005	23,203	3,867
Iteration 17	4,183	4,201	4,317	4,072	3,995	4,253	25,021	4,170
Iteration 18	4,271	4,137	4,458	4,248	3,866	4,120	25,100	4,183
Iteration 19	4,289	4,120	4,341	4,292	4,084	3,960	25,086	4,181
Iteration 20	3,818	3,168	4,192	4,106	3,413	3,410	22,107	3,685

Table 4.5 Simulated Aircraft Sortie Hours for Scenario 2

	Apr 43	May 43	Jun 43	Jul 43	Aug 43	Sep 43	Sum	Mean
Iteration 1	4,899	5,880	5,779	5,194	5,110	6,205	33,067	5,511
Iteration 2	6,494	6,086	5,932	6,141	4,829	5,211	34,693	5,782
Iteration 3	6,713	6,209	6,350	5,199	6,037	6,553	37,061	6,177
Iteration 4	6,979	6,994	6,743	6,354	5,725	6,605	39,400	6,567
Iteration 5	6,708	7,071	6,604	6,808	6,994	6,545	40,730	6,788
Iteration 6	6,543	6,965	6,502	6,724	6,915	6,540	40,189	6,698
Iteration 7	6,803	6,761	6,830	6,990	7,133	6,753	41,270	6,878
Iteration 8	6,849	6,926	6,462	6,705	7,260	6,879	41,081	6,847
Iteration 9	6,824	6,717	6,854	6,895	6,566	6,717	40,573	6,762
Iteration 10	7,080	7,026	6,673	6,735	6,941	6,541	40,996	6,833
Iteration 11	6,728	7,063	6,597	6,545	6,890	6,787	40,610	6,768
Iteration 12	6,907	7,132	6,894	7,102	7,018	6,759	41,812	6,969
Iteration 13	6,780	5,877	5,066	5,745	5,871	6,126	35,465	5,911
Iteration 14	5,827	5,744	5,684	6,347	6,338	6,526	36,466	6,078
Iteration 15	6,197	6,720	6,296	6,472	6,674	6,655	39,014	6,502
Iteration 16	6,321	6,825	6,674	6,267	6,965	6,693	39,745	6,624
Iteration 17	6,582	7,011	6,758	6,660	6,828	6,813	40,652	6,775
Iteration 18	6,486	6,913	6,618	7,073	6,963	6,867	40,920	6,820
Iteration 19	6,681	7,008	6,801	7,107	6,950	6,628	41,175	6,863
Iteration 20	6,952	7,043	6,697	6,913	7,053	6,621	41,279	6,880

In the following discussion, joint confidence interval bounds were computed using the t-distribution, given by

$$\bar{x} \pm \frac{s}{\sqrt{n}} t_{1-\frac{\alpha}{2k}, n-1} \qquad (4.20)$$

where $\bar{x}$ is the sample mean

s is the sample standard deviation

n is the sample size

k is the number of joint confidence intervals desired, and

$1-\frac{\alpha}{2k}$ is the joint confidence level desired with (n – 1) degrees of freedom.

Table 4.6 shows the total real-world sortie hours flown against the mean simulated totals (4.17) for Scenario 1 and Scenario 2. The confidence intervals were computed using (4.20) with 19 degrees of freedom and a joint confidence level of 0.8 (k = 2). Figure 4.15 depicts this data graphically.

Table 4.6 Total Sortie Hours, Simulated versus Actual

	Simulation Values			
Total Sortie Hours	Lower Conf. Bound	Sample Mean	Upper Conf. Bound	Actual
Scenario 1	22,362	23,189	24,016	24,230
Scenario 2	38,122	39,310	40,498	39,150

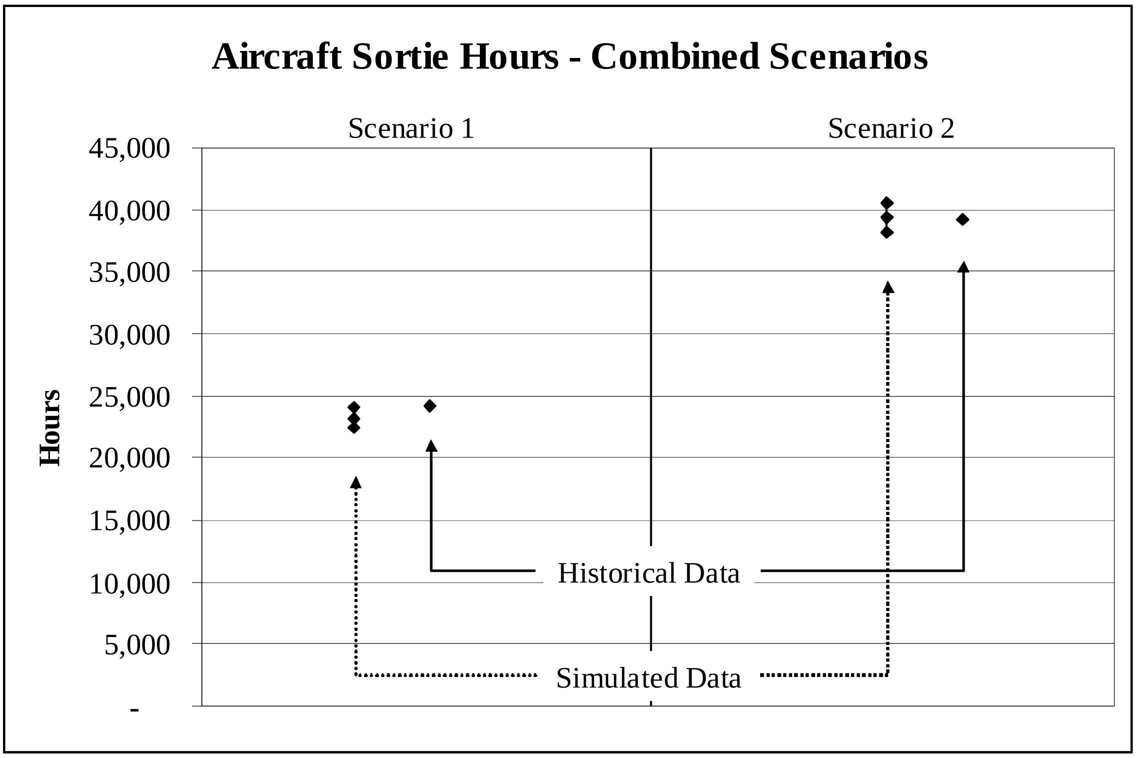

Figure 4.15 Total Sortie Hours Flown, Combined Scenarios

Given the aforementioned suspicions surrounding the accuracy of the historical record with respect to the sortie hours flown by the Allied aircraft, the sortie hours flown in each scenario were deemed sufficiently close to the actual data to represent a reasonably close level of effort for further MOE comparison. Indeed, the actual number of sortie hours for Scenario 1 is just outside the confidence interval by 214 hours, representing an average of 20 sorties per six months of simulation, or just over 1 extra sortie per aircraft per six months. The actual number of sortie hours for Scenario 2 is easily captured by the confidence interval. Thus, the model properly captures the Allied level of effort as measured by aircraft sortie hours.

4.4.2 Validation of Scenario 1 Results

Table 4.7 shows the simulation results for the number of U-Boat agents sighted by Allied aircraft agents during Scenario 1. The iterations' MOE totals accompany the monthly values, as do the monthly means.

Table 4.7 Simulated U-Boat Sightings for Scenario 1

	Oct 42	Nov 42	Dec 42	Jan 43	Feb 43	Mar 43	Sum	Mean
Iteration 1	9	17	21	17	11	33	108	18.000
Iteration 2	19	14	25	24	24	23	129	21.500
Iteration 3	16	23	15	22	25	28	129	21.500
Iteration 4	20	17	21	33	26	33	150	25.000
Iteration 5	15	16	18	25	28	26	128	21.333
Iteration 6	18	21	20	29	23	32	143	23.833
Iteration 7	11	20	24	30	34	28	147	24.500
Iteration 8	20	17	17	25	28	23	130	21.667
Iteration 9	27	25	34	40	28	30	184	30.667
Iteration 10	17	17	26	30	33	45	168	28.000
Iteration 11	9	9	23	13	21	27	102	17.000
Iteration 12	15	17	27	34	27	39	159	26.500
Iteration 13	12	14	18	21	17	25	107	17.833
Iteration 14	12	15	15	26	21	27	116	19.333
Iteration 15	13	17	16	24	25	36	131	21.833
Iteration 16	22	14	16	16	27	25	120	20.000
Iteration 17	21	15	23	17	21	23	120	20.000
Iteration 18	22	21	22	21	27	36	149	24.833
Iteration 19	21	28	32	30	24	21	156	26.000
Iteration 20	13	15	22	27	27	26	130	21.667

Table 4.8 shows the simulation results for the number of U-Boat agents destroyed by the Allied aircraft agents during Scenario 1. Like the previous table, the total number of kills and mean monthly kills accompany the raw monthly values.

Table 4.8 Simulated U-Boat Kills for Scenario 1

	Oct 42	Nov 42	Dec 42	Jan 43	Feb 43	Mar 43	Sum	Mean
Iteration 1	0	0	0	1	0	1	2	0.333
Iteration 2	0	0	1	1	2	1	5	0.833
Iteration 3	0	0	1	1	0	1	3	0.500
Iteration 4	0	0	1	0	1	1	3	0.500
Iteration 5	0	0	1	1	2	0	4	0.667
Iteration 6	0	0	0	1	1	0	2	0.333
Iteration 7	0	0	1	2	1	1	5	0.833
Iteration 8	0	1	0	0	1	1	3	0.500
Iteration 9	1	0	2	1	1	0	5	0.833
Iteration 10	1	1	2	1	1	0	6	1.000
Iteration 11	1	1	0	1	1	0	4	0.667
Iteration 12	1	0	1	0	1	0	3	0.500
Iteration 13	1	0	1	0	0	0	2	0.333
Iteration 14	0	0	0	0	1	1	2	0.333
Iteration 15	0	0	1	1	1	1	4	0.667
Iteration 16	2	1	0	0	1	0	4	0.667
Iteration 17	0	0	1	1	1	0	3	0.500
Iteration 18	0	0	2	1	0	2	5	0.833
Iteration 19	0	1	1	2	0	1	5	0.833
Iteration 20	0	1	1	0	1	1	4	0.667

Table 4.9 combines the MOE data from both the simulation and the historical record to facilitate comparison for validation.

Table 4.9 Combined MOEs for Scenario 1, Simulated versus Actual

	Simulation Values			
MOE	Lower Conf. Bound	Sample Mean	Upper Conf. Bound	Actual Data
Sightings	125.3	135.3	145.3	135.0
Kills	3.1	3.7	4.3	3.0

Figure 4.16 shows a graphical representation of this data. The confidence intervals have a joint confidence level of 0.8 (k = 2).

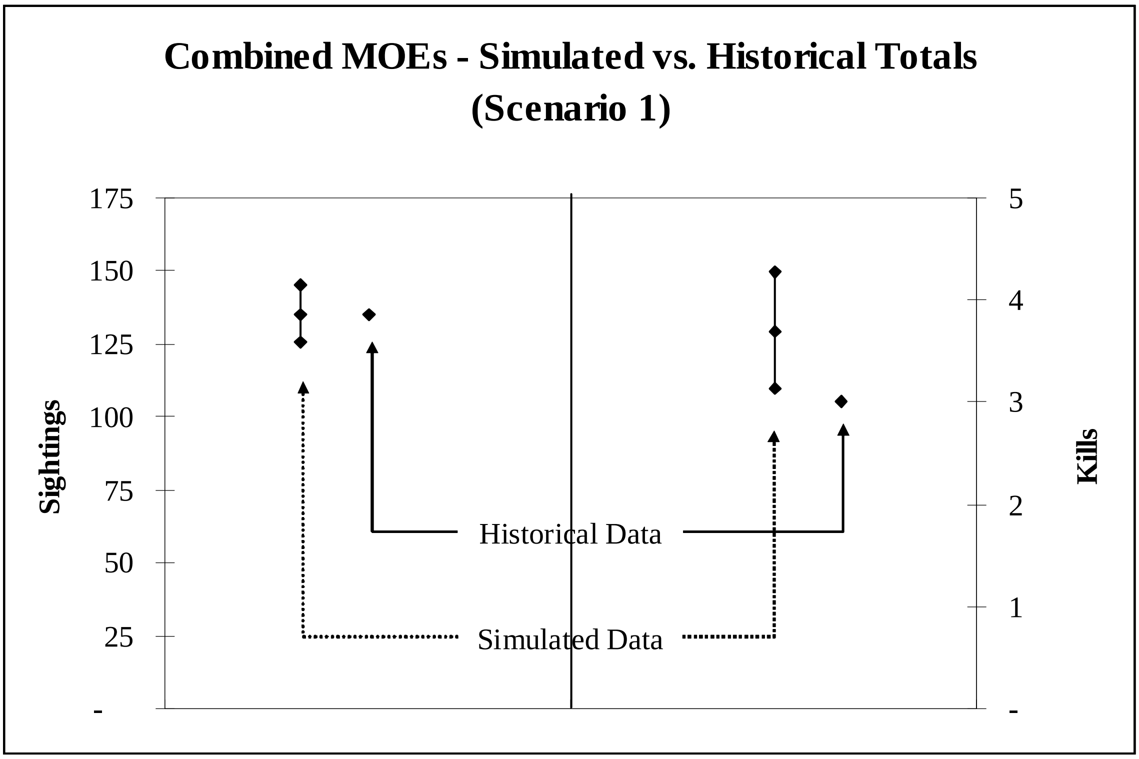

Figure 4.16 Comparisons of Simulated versus Historical MOE Values, Scenario 1

The Bay of Biscay agent-based simulation clearly produces an historically accurate number of U-Boat sightings. The number of U-Boats killed, however, falls slightly outside the confidence interval produced by the simulation results. The magnitude of the difference, however, is quite small and is less than a single kill over the 6-month scenario (indeed, the historical record is restricted to discrete integer values). Therefore, in spite of the statistical difference, it seems reasonable to say that the simulation produces accurate results for Scenario 1.

4.4.3 Validation of Scenario 2 Results

Table 4.10 shows the simulated sightings, and Table 4.11 shows the simulated kills for Scenario 2.

Table 4.10 Simulated U-Boat Sightings for Scenario 2

	Apr 43	May 43	Jun 43	Jul 43	Aug 43	Sep 43	Sum	Mean
Iteration 1	38	50	44	46	45	64	287	47.833
Iteration 2	48	46	49	57	62	70	332	55.333
Iteration 3	46	43	46	43	57	69	304	50.667
Iteration 4	46	48	51	56	69	48	318	53.000
Iteration 5	40	49	48	69	70	69	345	57.500
Iteration 6	60	46	67	70	58	57	358	59.667
Iteration 7	50	46	66	57	59	63	341	56.833
Iteration 8	42	52	46	54	74	79	347	57.833
Iteration 9	43	60	47	62	70	75	357	59.500
Iteration 10	46	53	54	72	75	73	373	62.167
Iteration 11	40	44	49	68	56	55	312	52.000
Iteration 12	36	59	51	67	63	58	334	55.667
Iteration 13	44	29	47	52	55	55	282	47.000
Iteration 14	35	40	49	45	71	48	288	48.000
Iteration 15	44	44	57	73	58	58	334	55.667
Iteration 16	42	58	54	61	60	68	343	57.167
Iteration 17	42	47	62	69	71	66	357	59.500
Iteration 18	43	59	56	79	74	65	376	62.667
Iteration 19	48	53	47	64	72	60	344	57.333
Iteration 20	41	45	57	61	59	75	338	56.333

Table 4.11 Simulated U-Boat Kills for Scenario 2

	Apr 43	May 43	Jun 43	Jul 43	Aug 43	Sep 43	Sum	Mean
Iteration 1	0	6	7	3	6	6	28	4.667
Iteration 2	1	3	4	8	5	5	26	4.333
Iteration 3	6	5	5	5	4	3	28	4.667
Iteration 4	2	9	4	3	9	3	30	5.000
Iteration 5	2	2	5	4	6	9	28	4.667
Iteration 6	4	5	8	8	8	5	38	6.333
Iteration 7	6	2	12	9	4	6	39	6.500
Iteration 8	3	2	8	8	9	13	43	7.167
Iteration 9	4	5	1	5	6	7	28	4.667
Iteration 10	5	4	4	6	13	5	37	6.167
Iteration 11	7	7	3	9	6	2	34	5.667
Iteration 12	6	3	2	12	9	5	37	6.167
Iteration 13	5	4	3	5	4	4	25	4.167
Iteration 14	2	4	7	2	8	4	27	4.500
Iteration 15	5	7	3	7	6	3	31	5.167
Iteration 16	6	6	6	3	5	11	37	6.167
Iteration 17	3	3	8	6	5	4	29	4.833
Iteration 18	2	6	5	6	5	6	30	5.000
Iteration 19	5	3	6	4	9	7	34	5.667
Iteration 20	3	7	4	6	5	7	32	5.333

Table 4.12 shows the statistical results in comparison to the historical outcome for Scenario 2. The confidence intervals have a joint confidence level of 0.8 (k = 2). The data are plotted in Figure 4.17.

Table 4.12 Combined MOEs for Scenario 2, Simulated versus Actual

	Simulation Values			
MOE	Lower Conf. Bound	Sample Mean	Upper Conf. Bound	Actual Data
Sightings	320.7	333.5	346.3	319.0
Kills	29.7	32.1	34.4	32.0

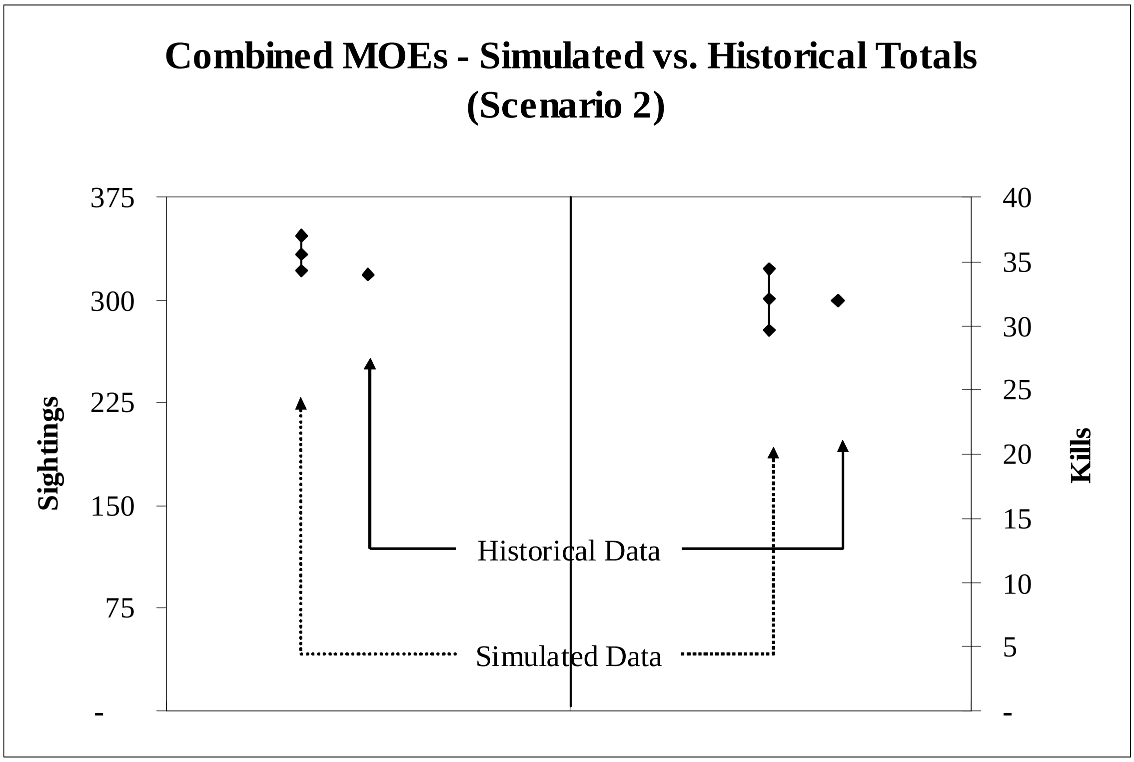

Figure 4.17 Comparisons of Simulated versus Historical MOE Values, Scenario 2

The simulation results for Scenario 2 compare very well to the historical outcomes. The confidence interval for simulated kills nicely encompasses the historical

number of kills for the same period. The confidence interval for sightings does not cover the historical value, but the magnitude of difference between the historical value and simulated mean is relatively small (within 5%). Indeed, the difference is 14.5 sightings over the 6-month scenario, an average of 2.4 sightings per month. Again, in spite of the statistical difference, it seems reasonable to say that the simulation produces accurate results for Scenario 2.

4.4.4 Validation of Emergent Behavior

While the validation of the simulation MOEs gives confidence as to the validity of the model, there are other characteristics that should be tested with agent-based models, specifically the emergent behavior of the agents themselves. For example, the operational analysts noted that the U-Boats entered the Bay of Biscay according to a Poisson distribution [Waddington, 1973; McCue, 1990]. The simulation model made no effort to force the U-Boat agents into specific behavior to conform to a Poisson arrival distribution. Therefore, the arrival times in the bay are an emergent phenomenon and can be statistically tested.

To test the arrival distribution, it is sufficient to recall that the inter-arrival times for a Poisson distribution are distributed exponential with parameter, λ. Gusella (1991) notes a common method for testing Poisson distributions of arrival processes in which the ratio of the mean to standard deviation of the inter-event times, called the index of dispersion, is calculated. The index of dispersion becomes the indicator, and for a Poisson process, the index of dispersion is equal to 1.

U-Boat arrival times were collected for each iteration during simulation execution, and the inter-arrival times were calculated. Table 4.13 shows the mean, variance, and their ratio for U-Boat inter-arrival times under each scenario.

Table 4.13 U-Boat Inter-arrival Statistics and Index of Dispersion

	Scenario 1			Scenario 2		
	Mean	St Dev	Ratio	Mean	St Dev	Ratio
Iteration 1	321.01	305.77	1.05	377.30	433.01	0.87
Iteration 2	352.23	346.22	1.02	394.56	383.23	1.03
Iteration 3	366.49	385.78	0.95	397.56	381.23	1.04
Iteration 4	372.62	372.94	1.00	369.04	444.17	0.83
Iteration 5	378.78	415.03	0.91	371.66	387.56	0.96
Iteration 6	402.23	455.46	0.88	372.11	419.52	0.89
Iteration 7	382.54	398.39	0.96	408.90	414.89	0.99
Iteration 8	371.38	407.55	0.91	398.49	492.90	0.81
Iteration 9	385.86	406.84	0.95	368.78	427.33	0.86
Iteration 10	402.35	396.20	1.02	384.38	436.37	0.88
Iteration 11	502.43	493.85	1.02	361.35	340.46	1.06
Iteration 12	412.88	389.28	1.06	419.88	389.28	1.08
Iteration 13	463.15	507.93	0.91	321.01	305.77	1.05
Iteration 14	390.02	401.04	0.97	352.23	346.22	1.02
Iteration 15	393.12	417.25	0.94	447.30	433.01	1.03
Iteration 16	361.35	340.46	1.06	406.06	380.03	1.07
Iteration 17	371.79	388.22	0.96	387.56	371.23	1.04
Iteration 18	419.71	401.51	1.05	379.24	423.16	0.90
Iteration 19	375.98	379.82	0.99	356.36	390.76	0.91
Iteration 20	366.82	390.46	0.94	362.13	413.82	0.88

Mean indexes of dispersion were computed for Scenario 1 and Scenario 2. Joint confidence intervals were constructed with a joint confidence level of 0.8 (k = 2). The results are displayed in Table 4.14.

Table 4.14 Index of Dispersion for U-Boat Inter-arrival Times

	Lower Bound	Mean	Upper Bound
Scenario 1	0.952	0.977	1.003
Scenario 2	0.918	0.960	1.002

The mean index of dispersion for both scenarios is very close to 1. In fact, the joint confidence intervals both cover 1.0, so there is not enough evidence to reject the hypothesis that the mean index is equal to 1. Therefore, the U-Boat arrival process appears Poisson distributed.

4.4.5 Validation Conclusions

By comparing the results of the Bay of Biscay agent-based simulation to the historical record, there are good indications that the model is capable of simulating the real-world scenario. The U-Boat arrival process in the simulation appears Poisson, as history indicates the real-world process was. Furthermore, given a level of effort of aircraft search sufficiently close to that in the real world, the simulation sightings and kills results are in line with the historical record. Though there are statistical differences in each scenario, the practical magnitude of these differences is relatively small. Given the model was able to produce similarly close results over two markedly different scenarios, the Bay of Biscay agent-based simulation is a robust representation as well.

There are two reasons for accepting the existing discrepancies between simulation and historical results. First, the real-world sortie hours are of suspect fidelity. As a result, the search effort can only be said to be "close" to the historical reality. Second, the statistical tests assume the real-world event represents the mean of all similar conflicts. The extent to which this particular conflict deviates from the mean of all such conflicts cannot be known, so exact validation tests (even statistical) are not achievable. However, a novel statistical approach for simulation validation of a mission-level model is developed in Chapter V. This test addresses the issue of uncertainty surrounding the

extent to which the single real-world outcome represents the mean result of all such conflicts.

As validation literature suggests, it is impossible to say with certainty that a model is validated. However, the statistical validation tests outlined above indicate the Bay of Biscay agent-based simulation is a good representation of the real-world operation.

4.5 Extensions to Modern Problems

In the more than sixty years since the Bay of Biscay campaign took place, submarine technology has outdated the simulated scenarios with respect to modern submarine/anti-submarine operations. Modern nuclear submarines are faster, do not need to surface for extraordinarily extended periods of time, are able to stay out of port for months of continuous operation, and are able to travel much deeper than was possible during WW II. As a result, radar and visual search by air for submarines is generally an ineffective proposition. In spite of this, the basis for the Bay of Biscay scenario can be widely applied to current operations, beyond purely military applications and into the realm of law enforcement, immigration, treaty verification, arms inspection, and others.

4.5.1 Scenario Fundamentals

The properties underlying the offensive search for U-Boats in the Bay of Biscay suggest that other situations may be investigated with similar agent-based tools. Because of the nature of these situations, the discussion is from the viewpoint of the searching party.

One of the primary characteristics of the Bay of Biscay scenario is that the target may not be in the search zone. Fundamentally, this aspect varies from the majority of the modern literature on analytical search methods, which typically assume one or more targets within the search zone. Though the target is known to pass through the search area, there are an unknown number of targets in the region at any given time.

Although the area of origin and area of operation are well known to the searchers, these areas are beyond their influence, so action against the targets is severely constrained at the point of origin and operation and is effectively possible only when the target is in transit between its origin and operational zone. Moreover, it is known that the target must pass through the search zone to get to its operating zone, and it must pass through it again on its way back to its origin point.

The target is mobile, and while in transit, the target is uncooperative (in search terminology this means the target is not willing to be found and is actively working to avoid detection). However, while the target is uncooperative, it is visible and vulnerable to detection, at least for short time periods.

Finally, the search assets come from outside the search area. These assets are limited in number and capability, and as a result, they are not always in the search zone.

4.5.2 Possible Modern Applications

Though this research may no longer be applicable to anti-submarine operations, there are modern applications which have characteristics similar to the simulated scenario. Several of these are discussed below.

Illegal Immigration: Border control is an important issue that has many characteristics featured in the Bay of Biscay agent-based simulation. The illegal immigrants (targets) leave their country (point of origin), cross a border (search zone), and eventually meld into the population of the destination country (operation zone). The border patrol has limited assets and must cover a lengthy border. The point of origin is outside the jurisdiction of the border patrol (searchers), and once mingled with the host population, the targets are very difficult to identify. However, there is a time between crossing the border and reaching their destination when targets are vulnerable to detection.

(Drug) Smuggling Interdiction: Smuggling scenarios are very similar to scenarios involving illegal immigration, and the smuggling of drugs from one country to another is of particular concern. The drug smugglers (targets) leave their country (point of origin) with the product, cross a border (search zone), and eventually deliver to the front end of some domestic distribution system (operation zone). Once the product enters the distribution system, it becomes very difficult to effectively interdict, and the country of origin is outside the direct control of the searchers. However, interdiction in transit, when the product is massed, provides the opportunity to effectively impact illicit product supply in the operation zone.

Terrorist Identification and Interdiction: Terrorist identification and interdiction is a subject currently gaining an enormous amount of attention, and it is a scenario to which this type of agent-based simulation may be able to provide significant insights. Since terrorists most often do not wear uniforms, they are not visible as terrorists until

they are in the process of a terrorist act. Once they reach their operation zone, it is often too late to prevent their mission from being at least partially carried out. Therefore, the opportunity to identify and interdict them must be while in transit. This is perhaps most applicable to the Israelis, who share a controlled border with the typical terrorist population.

Treaty Verification (SCUD Hunting): Though SCUD hunting differs somewhat from the previous examples, it is sufficiently similar to indicate that agent-based simulation may be applicable. In the case of a banned, but deployable, weapon system such as the SCUD missile in Iraq, the weapon system can be hidden or made to blend in with other equipment, but when deployed, the system is vulnerable to detection. Since the system has a limited range, search can be limited to areas from where launches would most likely occur to strike probable targets. Again, limited search assets are available and must be mobile to “catch” the system when it is deployed.

This application is particularly interesting in the context of the Bay of Biscay agent-based simulation as well. Finding SCUD missiles has been a significant political objective since the Gulf War, and as a result, it has received a considerable amount of consideration within the military community. The notion of applying anti-submarine warfare (ASW) principles to finding SCUD missiles was proposed in [Wirtz, 1997] and [Connor, 1997], and successful application to ASW in the Bay of Biscay agent-based simulation suggests that the techniques of agent-based simulation could be extended to the problem of locating SCUD missiles as well.

Mobile Chemical Weapons Production Facilities: Like the previous example, searching for mobile chemical and biological weapons production facilities is a scenario that differs somewhat from that of the Bay of Biscay, but it does have enough similarities to indicate an agent-based approach may provide insights.

Mobile chemical weapons production facilities are virtually impossible to find and identify when not in production mode. However, when producing the chemical or biological agents, the facility must be stationary. Moreover, specific, easily identifiable support equipment must be present when production of the chemical agents is ongoing. Therefore, while in production mode, the facility is vulnerable to detection. Additionally, these facilities must be within range of delivering their products to capable handling facilities [Powell, 2003]. Therefore, a probable search area can be determined for extremely limited search assets within a hostile environment.

4.5.3 Summary

Though the preceding examples are not the only scenarios that have the above characteristics, these are some that are directly concerned with national security and have been of recent widespread interest.

4.6 Conclusion

This chapter outlined several important contributions to the field of agent-based combat simulation. First, through the development of the Bay of Biscay agent-based simulation, the state-of-the-art in agent-based combat simulation is extended. This simulation is the first agent-based combat simulation to reproduce a real-world mission-

level scenario. Second, the simulation was validated against the historical record, including the emergent behavior of the U-Boat agents. Third, through the validation of the simulation, a use for the V&V taxonomy outlined in Chapter III was demonstrated. Fourth, acknowledging the remoteness of the simulation to modern anti-submarine activity, the Bay of Biscay offensive search scenario was tied to relevant modern security/defense applications.

V. New Statistical Approach to Validating Agent-Based Combat Simulations

Combat, unlike many real-world processes, tends to be singular in nature. That is, there are not multiple occurrences from which to hypothesize a probability distribution model of the real-world system. Engagement models tend to be singular due to their relatively short duration. Mission-level models may offer more flexibility on some measures due to their extended time frame. Additionally, the parameters involved in the model may be unchanged for significant stretches of the total simulation time. In these cases, time periods may be devised such that the periods hold sufficiently similar traits such that the incremental results may be assumed to come from a common distribution. For example, with respect to a simulation modeling several months of operations, the results may be compiled monthly, thereby providing multiple samples of historic behavior from a single instance.

This chapter details a new statistical test for use in validating a mission-level model. The test is developed within the context of the Bay of Biscay agent-based simulation and uses the monthly data from the extended campaign as a basis of comparison to the simulation output.

5.1 Motivation for a New Validation Test

In the previous chapter, several standard statistical tests for the validation of a combat simulation were presented. The tests compared the overall MOE values for each

of two scenarios simulating six months of combat operations. The comparisons between historical and simulated outcomes were favorable, and the validation process suggests that the simulation is a good representation of the scenarios as they happened. The fact remains, however, that the historical outcome is itself a single sample from a stochastic process (i.e. combat). The statistical comparisons made in the validation process were based on the assumption that the historic results represent the mean value of all possible outcomes. A favorable comparison of the simulation with the underlying stochastic process that produced the single historic sample would provide greater confidence that the model is a valid representation of the real-world system.

Examining Bay of Biscay historic outcomes by month, instead of aggregated, provides a convenient method for examining the variability of the real-world system. Mean monthly values for each MOE of interest (4.18), both real-world and simulated, can be calculated and compared. The resulting analysis provides additional insight not available through the techniques previously presented, although it still lacks quantifiable confidence to conclusions about the validity of the simulation. The data generated from the Bay of Biscay agent-based simulation are used to demonstrate the strengths and weaknesses of this approach.

Figures 5.1 through 5.6 depict the historic versus simulated mean monthly MOE values via joint confidence intervals for each of the three MOEs in both scenarios. Each figure shows 21 individual confidence intervals – the left-most being the historic value with the remaining 20 coming from each of 20 simulation iterations. Joint confidence

intervals were constructed to allow an overall 80% joint confidence level (k = 2) for each comparison.

Figures 5.1 and 5.2 show the mean monthly aircraft sortie hours for Scenario 1 and Scenario 2, respectively.

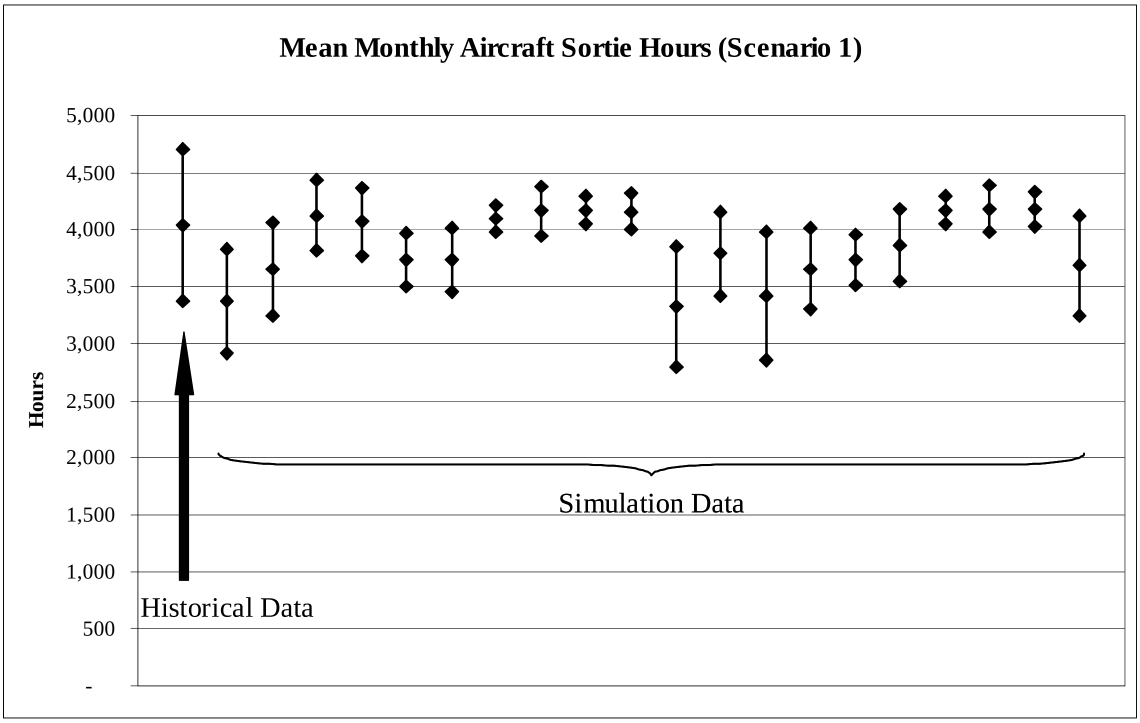

Figure 5.1 Comparisons of Mean Monthly Sortie Hours, Historic vs. Simulated Scenario 1

The confidence intervals from each simulation iteration of Scenario 1 overlap the confidence interval derived from the historical data. The implication from this is that for any individual comparison between an iteration and the historical data, the means cannot be said to be statistically (significantly) different.

Recall from Table 4.6, the real-world sortie hour total over the 6-month scenario was slightly outside the confidence interval generated from the simulation iteration totals. The validation argument used to accept the result as valid despite the difference was based on the uncertainty surrounding the veracity of the real-world records and the small magnitude of the difference when viewed in practical terms. The results demonstrated in Figure 5.1 reinforce this conclusion.

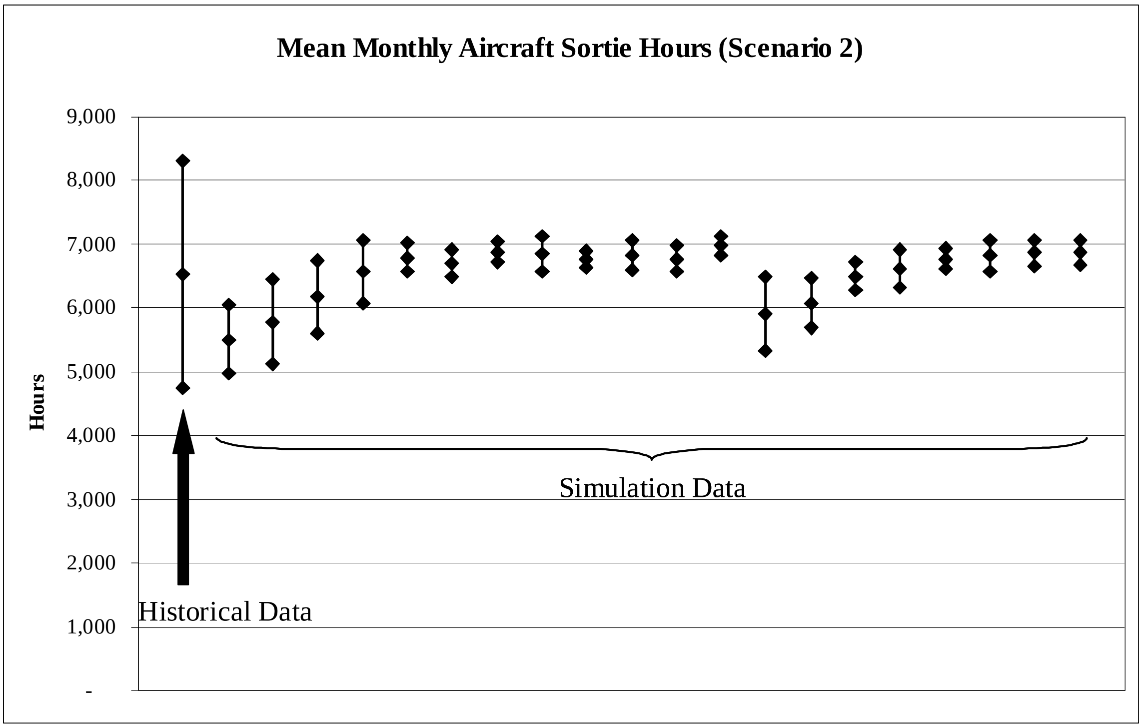

Figure 5.2 Comparisons of Mean Monthly Sortie Hours, Historic vs. Simulated Scenario 2

In Figure 5.2, all the confidence intervals derived from the output data overlap the confidence interval derived from the historical data for Scenario 2. As with Figure 5.1, Figure 5.2 gives face-level support (with no real statistical confidence added) that the

level of effort for Allied aircraft agents within the simulation is a reasonable approximation for the actual level of effort used in the real-world operation.

Figure 5.1 and Figure 5.2 indicate that variance of the simulation is apparently smaller than that of the real-world process by an appreciable amount. This is an expected result in the case of sortie hours flown. One reason for this is that weather is one of the two stochastic factors controlling sortie generation. The simulated impact of weather is a probability derived as an average value of sorties cancelled over the entire four years of operations in the Bay of Biscay. This averaging smoothed the variation that actually occurred month-to-month. Scenario 2 is particularly impacted by this because the summer of 1943 had unusually good weather. As a result of good weather, the Allied aircraft were able to fly an unusually large percentage of scheduled sorties [McCue, 1990].

Even with no further analysis, a major shortcoming of this validation approach becomes evident. In preparing for the comparisons, an analyst must choose two unattractive options when constructing joint confidence intervals. The first option is to compare each simulation iteration to the historic data at some known confidence level (e.g. 80% with $k = 2$, as presented in Figures 5.1 through 5.6). The second option is to construct the intervals such that all simulation iterations versus historic outcome comparisons taken together have a known joint confidence level (i.e. $k = 21$). If the former option is chosen, the resulting joint confidence level for all 20 comparisons is near zero. If the latter is chosen, the overall confidence level is known, but the individual confidence intervals are so large they cease to be discriminating.

Acknowledging this significant shortfall in the approach, the MOE results (U-Boats sighted and U-Boats killed) for each of the simulated scenarios are presented. Figure 5.3 and Figure 5.4 show the comparison of results from Scenario 1.

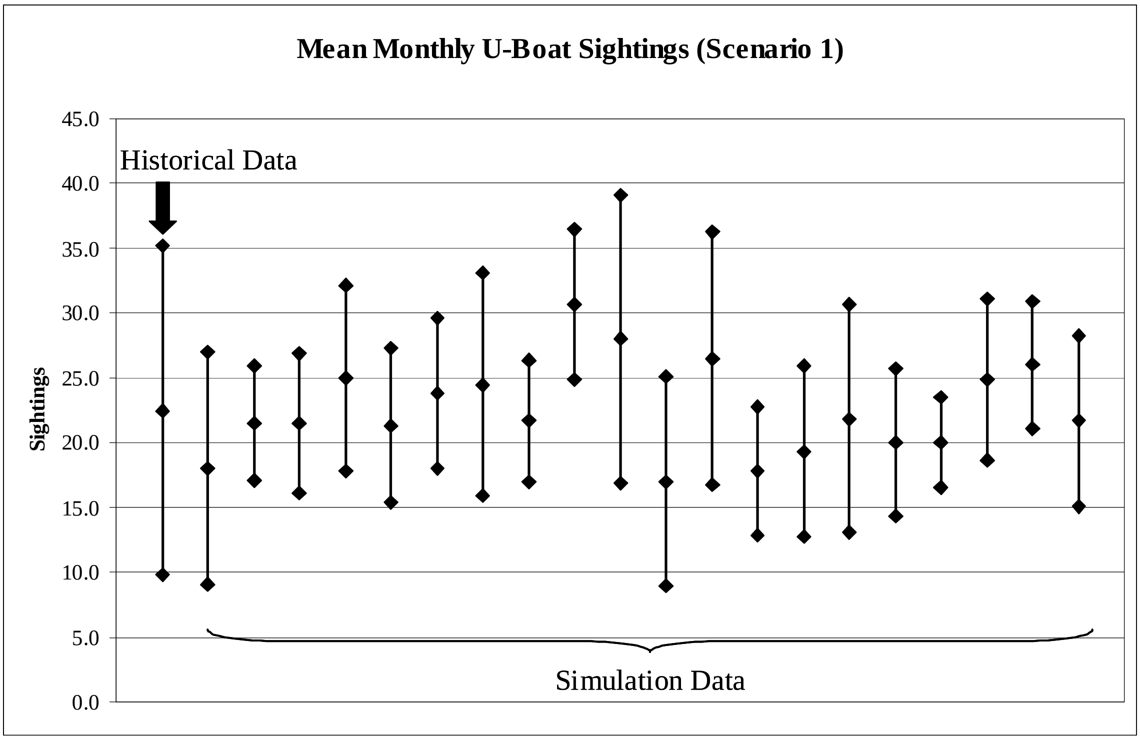

Figure 5.3 Comparisons of Mean Monthly U-Boat Sightings, Historic vs. Simulated Scenario 1

In Figure 5.3, as with the comparisons of levels of effort, there is 100% overlap of the confidence intervals generated from the mean monthly U-Boat sightings and the confidence interval derived from the historical data.

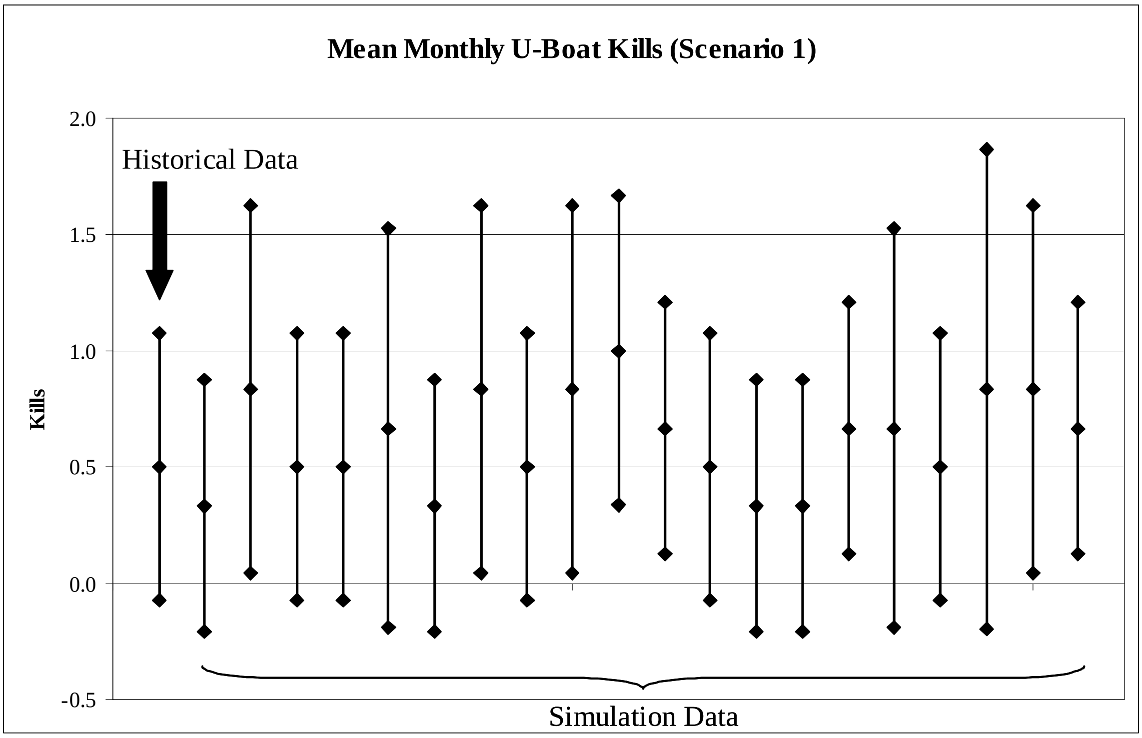

Figure 5.4 Comparisons of Mean Monthly U-Boat Kills, Historic vs. Simulated Scenario 1

In Figure 5.4, there is again 100% overlap of confidence intervals in comparing individual simulation iteration means to the real-world data. Recall from Table 4.9 that the number of total kills over the 6-month scenario fell slightly outside the confidence interval derived from the simulation totals. As with the case of total sortie hours, the practical implications of the difference were small, and the simulated result was accepted as a valid approximation of the real-world system. Figure 5.4 provides face-level support for this conclusion.

Figure 5.5 and Figure 5.6 show the simulation results for U-Boats sighted and U-Boats killed, respectively, for Scenario 2.

Figure 5.5 Comparisons of Mean Monthly U-Boat Sightings, Historic vs. Simulated Scenario 2

As in the previous examples, Figure 5.5 indicates 100% confidence level overlap in comparing individual simulation iteration means to the real-world data. Recall from Table 4.12 that the number of total U-Boat sightings over the 6-month scenario fell slightly outside the confidence interval derived from the simulation totals. As with the case of total sortie hours and U-Boat kills in Scenario 1, the practical implication of this difference was small, and the simulated result was accepted as a valid approximation of the real-world system. Figure 5.5 provides face-level support for this conclusion.

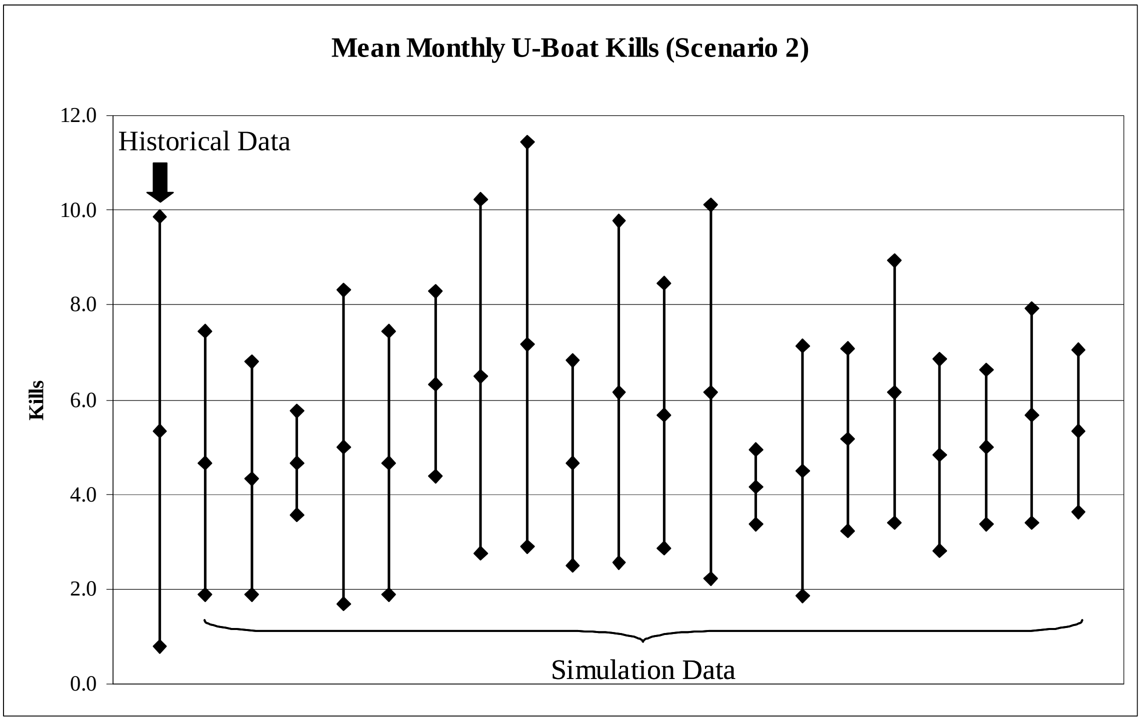

Figure 5.6 Comparisons of Mean Monthly U-Boat Kills, Historic vs. Simulated Scenario 2

Figure 5.6 demonstrates that there is 100% overlap of the confidence intervals generated from the mean monthly U-Boat kills and that of the confidence interval derived from the real-world data for Scenario 2.

Because of the analytic dilemma surrounding the joint confidence level, this method of analysis provides little more than face-level confidence. The statistical confidence remains near zero. However, the approach is tempting in that it offers insight into the stochastic nature underlying a real-world system with a single occurrence (sample size of 1). The remainder of this chapter is devoted to developing and demonstrating a test methodology that allows for statistically significant comparisons, despite having a single real-world sample.

5.2 Methodology for Comparison of Historic versus Simulated Data

Any test allowing a meaningful comparison between the historic outcome and the simulated data, while still providing insight into the underlying stochastic real-world system, requires two characteristics. First, the method must provide a means of deriving multiple samples from the stochastic process underlying the real-world system. Second, the method must provide a meaningful, quantifiable level of confidence in the result. Figure 5.7 illustrates an approach that meets both requirements.

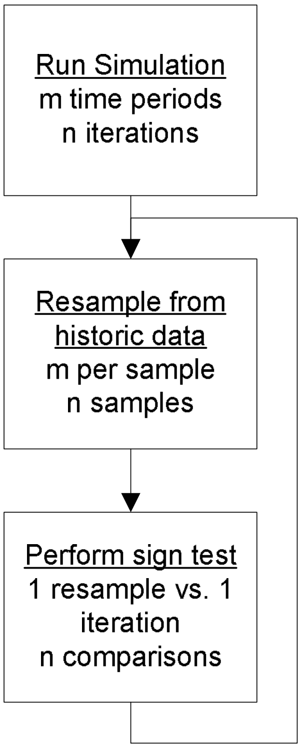

Figure 5.7 Methodology for Comparisons of a Single-Sampled Real-World Process to Simulated Results

Once the simulation results from *n* iterations are generated, the historic data is used to generate *n* bootstrap samples. A sign test is used to test the hypothesis that the

two samples are statistically identical. The bootstrap and sign test is then replicated for multiple experiments.

The basic approach above is based on well-accepted nonparametric statistical techniques. Once the simulation data has been collected, the basic approach has the added benefit of being simple to execute and can be quickly performed within a spreadsheet.

5.2.1 Bootstrap

Several statistical resampling techniques have been developed to provide estimators of population parameters that are difficult or impossible to treat theoretically [Conover, 1999] or when obtaining multiple samples from a system is prohibitively expensive [Cheng, 2001]. Resampling is based on the idea that when one random sample is available and obtaining another sample is not feasible, then the best estimate for the distribution under study is the random sample in-hand.

Efron (1979) first proposed the bootstrap method of resampling. Since it was first proposed, the method has found wide acceptance and applicability. Efron and Tibshirani (1986) review the bootstrap method and its applications.

<u>The Method</u>: Consider the statistic θ calculated from the random sample X = {X_1, X_2, …, X_n}. A bootstrap sample X^* = { X_1^*, X_2^*, …, X_n^*} is generated by taking a random sample from X, where $P(X_j^*(j=1, 2,...,n) = X_i(i=1, 2,...,n)) = \frac{1}{n}$, for which θ^*, an estimator for θ, is computed from the bootstrap sample. If some number, B, Monte

Carlo replications are taken, then the distribution of θ can be estimated by the sample mean and standard deviation of θ^*.

Sample Size, B: The number of bootstrap samples needed to accurately estimate the properties of the sample statistic vary. Efron and Tibshirani (1986) note that for most situations, B = 50 to 200 is "quite adequate," though 250 or more are often needed for accurate computation of confidence intervals. Conover (1999) adds that "as few as 25 replications can be very informative".

Proposed Use: The bootstrap used differs slightly for the proposed methodology. Instead of a single collection of bootstrap samples of the historic data, m groups of b bootstrap samples were generated for comparison with the simulation, where b = the number of simulation iterations and m = number of sign test trials desired.

Assumptions and Remedial Methods: Bootstrap resampling assumes the original sample is independent and identically distributed (i.i.d.). Since the historic data from the Bay of Biscay operations consists of calendar data (i.e. time-series data), it is likely that the MOE data is autocorrelated to some degree. Table 5.1 shows the calculated autocorrelation (1 time lag) for the data from each Scenario.

Table 5.1 Autocorrelation of Historic MOE Values

	Scenario 1	Scenario 2
Sortie Hours	0.0688	0.4732
U-Boat Sightings	0.5345	0.1192
U-Boat Kills	0.1667	-0.3189

From Table 5.1, it appears that autocorrelation is an issue with Scenario 1 U-Boat Sightings, Scenario 2 Sortie Hours, and Scenario 2 U-Boat kills. Statistically, however, the extremely small sample size (n = 6) for both Scenarios does not provide any conclusive evidence that the samples are autocorrelated. This small sample size also prevents the practical application of remedial data measures that could treat the correlation within the samples. There are methods of treating autocorrelated samples so that the bootstrap assumptions can be met. The moving blocks bootstrap is one method that extends the bootstrap to time series data [Dixon, 2001].

In the moving blocks bootstrap, the time series data is partitioned into b non-overlapping blocks consisting of l sequential observations. Values of b and l are chosen so that the correlation within each of the blocks is strong, but weak between blocks. With l correctly chosen, the b blocks are considered independent. The bootstrap method randomly samples with replacement from the b blocks to obtain a series of $b \cdot l$ observations.

The moving blocks bootstrap is not a feasible solution to the specific problem posed by the Bay of Biscay scenario validation data. The small number of observations in each validation set prevents effective blocking schemes. The fidelity of the available data also represents an obstacle. Data for the Bay of Biscay operations are available in monthly increments (observations). If the data were available in smaller time increments (more observations), then perhaps a viable blocking scheme could be contrived.

Combat operations will perpetually pose sample size problems since real-world operations seldom maintain stationary/static strategies, tactics, or technologies long enough to produce data of a significant sample size.

5.2.2 Sign Test

The sign test is used to test whether one random variable in a pair (X, Y) tends to be larger than the other random variable in the pair. It is a variant of the binomial test in which the probability of outcome is assumed to be equally likely, $p = 1 - p = 0.5$ [Conover, 1999].

Data for the sign test consists of *n'* pairs of observations (X_1, Y_1), (X_2, Y_2), …, $(X_{n'}, Y_{n'})$, each observation being a bivariate random sample. Within each (X_i, Y_i) observation, a comparison is made, and the pair is classified as "+" if $X_i < Y_i$, "–" if $X_i > Y_i$, or "0" if $X_i = Y_i$. The test statistic, *T*, is the number of "+" pairs. The null distribution of *T* is the binomial distribution with $p = ½$ and n = number of non-tied pairs (tied pairs are disregarded).

The sign test assumes that the bivariate pairs are mutually independent, and the probability of outcome is constant for all trials. It further assumes that the measurement scale within each pair is at least ordinal, that is each (X_i, Y_i) pair may be determined to be "+", "–", or "0". Finally, the sign test assumes there is internal consistency between the observed pairs.

For model validation purposes, the two-tailed test is desired. That is,

$$H_0: P(+) = P(-)$$

$$H_1: P(+) \neq P(-).$$

This is the hypothesis test demonstrated with the Bay of Biscay agent-based simulation data in Section 5.3.

The critical α-values are determined for each test once n has been determined. Because the binomial distribution is discrete, the critical α-values cannot be arbitrarily set. Instead, the critical α-level is selected such that the total $(1 - \alpha)$ level is as close to 0.9 as possible, without being less than 0.9, given a particular n. That is, H_0 is rejected if the p-value for the test is less than 0.05.

5.3 Bay of Biscay Agent-Based Simulation Results

The presentation of results follows the same order as in the previous analyses. That is, the comparisons of sortie hours for both scenarios are presented first, followed by the remaining MOEs from each scenario, respectively.

Each MOE was subjected to identical experiments. Each experiment consists of twenty sign tests ($m = 20$), with each sign test incorporating twenty (one per simulation iteration) bootstrap samples ($b = 20$). For each MOE, one sign test is presented in detail, and the remaining tests are summarized prior to validation discussions.

5.3.1 Sortie Hours

Previous analyses of sortie hours provided a somewhat mixed picture of the simulation's fidelity with respect to the historic data. The historic sortie hour total for Scenario 1 was slightly outside the simulation confidence interval, though the practical

difference was negligible. Comparisons between the confidence interval generated by the historic monthly data and those generated from each iteration's monthly data, however, demonstrated 100% overlap, and hence, no statistical difference between the results from any individual iteration and the historic data. This approach, however, lacked any meaningful confidence when all such comparisons were taken together. The historic sortie hour total from Scenario 2 was well within the confidence interval derived from the simulation data.

Table 5.2 shows the bootstrap samples for Scenario 1 sortie hours generated for a single replication of the bootstrap/sign test experiment. The monthly bootstrap sortie hours are totaled in the right-most column.

Table 5.2 Bootstrap Sortie Hours – Scenario 1

	Sortie Hours						
Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,400	3,400	4,100	4,600	4,100	3,130	22,730
2	3,400	3,130	4,100	3,130	4,400	4,600	22,760
3	3,400	3,400	4,100	4,400	4,600	4,600	24,500
4	4,100	4,600	4,600	4,400	4,600	3,130	25,430
5	4,600	4,600	4,100	4,100	3,400	4,100	24,900
6	4,600	4,100	4,400	4,100	4,600	4,400	26,200
7	4,600	3,400	4,100	4,400	4,100	3,130	23,730
8	4,100	4,400	4,100	4,400	4,600	3,400	25,000
9	3,130	4,400	4,400	4,600	4,400	4,600	25,530
10	3,130	3,130	4,400	4,100	4,400	4,400	23,560
11	3,400	4,100	4,100	4,600	4,100	4,600	24,900
12	4,100	4,600	4,100	4,100	4,100	4,600	25,600
13	3,130	4,400	3,130	4,100	4,600	4,100	23,460
14	3,130	3,400	4,600	4,400	4,600	4,100	24,230
15	4,600	4,600	3,130	3,400	3,130	3,130	21,990
16	3,400	4,100	4,400	3,130	3,130	4,100	22,260
17	3,130	4,600	3,130	3,130	4,100	4,100	22,190
18	3,400	4,600	3,130	4,400	4,100	4,600	24,230
19	3,400	3,400	4,400	4,600	4,600	4,600	25,000
20	4,400	4,600	4,600	4,600	4,600	3,130	25,930

Table 5.3 summarizes the sign test classifications for the paired data (X_i, Y_i) for Scenario 1, where X_i is the i^{th} bootstrap sortie hour total and Y_i is the sortie hour total from the i^{th} simulation iteration from Table 4.4. The sign test statistic T and number of non-tied pairs n are displayed as well.

Table 5.3 Sign Test Calculations – Sortie Hours, Scenario 1

Observation	1	2	3	4	5	6	7	8	9	10
Sign	–	–	+	–	–	–	+	–	–	+
Observation	11	12	13	14	15	16	17	18	19	20
Sign	–	–	–	–	+	+	+	+	+	–
T	8									
n	20									

For $n = 20$, $P(t \leq 5) = 0.0207$ and $P(t \geq 14) = 0.0207$ defining an overall $(1 - \alpha) = 0.9586$. Since $5 < T = 8 < 14$, there is insufficient evidence to reject H_0. For this trial, there is no compelling evidence to suggest the simulation does not faithfully represent the real-world system with respect to Scenario 1 sortie hours. The resulting p-value is 0.2517.

The results for the entire experiment are summarized in Table 5.4. Of the 20 sign test trials, the p-values ranged in value from 0.021 to 0.412. Under the rejection criteria, the null hypothesis was rejected for 3 of the 20 trials.

Table 5.4 Summary of 20 Bootstrap Experiments for Scenario 1 Sortie Hours

	Comparison Classification																					
Trial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	T	n	p–value
1	–	–	+	–	–	–	+	–	–	+	–	–	–	–	+	+	+	+	+	–	8	20	0.252
2	–	–	–	+	–	–	+	+	+	+	–	–	–	–	–	+	+	+	–	–	8	20	0.252
3	–	–	–	+	–	–	+	–	+	+	–	–	–	–	–	–	–	+	+	–	6	20	0.058
4	–	–	+	+	–	–	–	+	+	–	–	–	–	–	–	–	–	+	+	–	6	20	0.058
5	–	–	+	+	–	–	+	+	+	+	–	–	–	–	–	+	–	+	–	–	8	20	0.252
6	–	–	+	–	–	–	–	–	–	+	–	–	–	–	–	–	+	+	+	–	5	20	0.021
7	–	–	+	+	–	–	–	–	+	+	–	–	–	–	–	–	+	–	+	–	6	20	0.058
8	–	–	+	+	–	–	–	+	+	+	–	–	–	–	–	+	+	+	+	–	9	20	0.412
9	–	+	–	+	–	–	–	+	+	+	–	+	–	–	+	–	+	–	–	–	8	20	0.252
10	–	–	+	+	–	–	+	+	+	+	–	–	–	–	–	–	+	+	+	–	9	20	0.412
11	–	–	+	+	–	–	–	+	+	+	–	–	–	–	–	+	+	+	+	–	9	20	0.412
12	–	–	+	+	–	+	–	–	+	–	–	–	–	–	–	–	–	–	+	–	5	20	0.021
13	–	–	+	–	–	–	+	–	–	–	–	–	–	+	–	–	–	–	+	+	5	20	0.021
14	–	–	–	+	–	–	+	+	+	+	–	–	–	–	–	–	+	+	+	–	8	20	0.252
15	–	–	+	+	–	–	–	+	–	+	–	–	–	–	–	–	+	+	+	–	7	20	0.132
16	–	–	+	+	–	+	–	–	+	+	–	–	–	–	–	–	+	–	–	–	6	20	0.058
17	–	–	+	+	–	–	–	+	+	+	–	–	–	–	–	–	+	–	+	–	7	20	0.132
18	–	+	–	+	–	–	+	–	+	–	–	–	–	–	–	–	+	+	+	–	7	20	0.132
19	–	–	+	+	–	–	–	+	+	–	–	–	–	–	–	–	+	+	+	–	7	20	0.132
20	–	–	+	–	–	–	+	+	+	+	–	–	–	–	+	–	–	–	+	–	7	20	0.132

Table 5.5 shows the bootstrap samples of Scenario 2 sortie hours generated for a single replication of the bootstrap/sign test experiment. The monthly bootstrap sortie hours are totaled in the right-most column.

Table 5.5 Bootstrap Sortie Hours – Scenario 2

	Sortie Hours						
Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5,900	5,350	7,000	5,900	7,000	7,000	38,150
2	8,000	8,700	8,700	7,000	5,900	7,000	45,300
3	5,900	5,900	4,200	4,200	7,000	8,700	35,900
4	7,000	8,000	5,900	4,200	8,000	8,700	41,800
5	8,000	4,200	8,700	5,900	5,350	4,200	36,350
6	7,000	5,900	7,000	8,000	4,200	5,900	38,000
7	7,000	7,000	7,000	7,000	4,200	7,000	39,200
8	5,350	5,350	8,700	5,350	5,900	5,350	36,000
9	4,200	5,350	7,000	8,700	5,350	5,350	35,950
10	7,000	8,000	7,000	8,700	8,700	7,000	46,400
11	8,000	5,350	8,700	7,000	8,700	5,350	43,100
12	5,350	8,700	5,900	8,000	4,200	7,000	39,150
13	8,700	8,000	5,350	8,000	5,900	4,200	40,150
14	4,200	8,700	5,350	7,000	5,900	5,900	37,050
15	8,700	8,000	5,350	5,900	4,200	8,700	40,850
16	8,700	5,350	7,000	8,700	5,900	5,350	41,000
17	8,700	5,900	4,200	5,350	8,700	8,000	40,850
18	4,200	4,200	5,350	8,700	8,700	8,700	39,850
19	5,900	7,000	7,000	5,350	8,700	5,350	39,300
20	4,200	7,000	8,000	8,700	5,350	4,200	37,450

Table 5.6 summarizes the sign test classifications for the paired data (X_i, Y_i) for Scenario 2 sortie hours, where X_i is the i^{th} bootstrap sortie hour total and Y_i is the sortie hour total from the i^{th} simulation iteration from Table 4.5. The sign test statistic T and number of non-tied pairs n are displayed as well.

Table 5.6 Sign Test Calculations – Sortie Hours, Scenario 2

Observation	1	2	3	4	5	6	7	8	9	10
Sign	–	–	+	–	+	+	+	+	+	–
Observation	11	12	13	14	15	16	17	18	19	20
Sign	–	+	–	–	–	–	–	+	+	+
T	10									
n	20									

For $n = 20$, $P(t \leq 5) = 0.0207$ and $P(t \geq 14) = 0.0207$ defining an overall $(1 - \alpha) = 0.9586$. Since $5 < T = 10 < 14$, there is insufficient evidence to reject H_0. There is no compelling evidence to suggest the simulation does not faithfully represent the real-world system with respect to Scenario 2 sortie hours. The resulting p-value is 0.3238.

The results for the entire experiment are summarized in Table 5.7. Of the 20 sign test trials, the p-values ranged in value from 0.021 to 0.412. Under the rejection criteria, the null hypothesis was rejected in 1 of the 20 trials.

Table 5.7 Summary of 20 Bootstrap Experiments for Scenario 2 Sortie Hours

	Comparison Classification																						
Trial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	T	n	p–value
1	–	–	+	–	+	+	+	+	+	–	–	+	–	–	–	–	–	+	+	+	10	20	0.412
2	–	+	–	–	+	–	+	+	+	+	+	+	–	–	–	+	+	+	+	+	13	20	0.058
3	–	–	–	+	+	+	+	+	–	+	+	+	–	–	+	–	+	+	+	–	12	20	0.132
4	–	–	–	–	+	–	+	+	+	–	+	+	+	+	–	+	+	+	+	+	13	20	0.058
5	–	–	+	+	+	–	+	+	+	–	–	+	–	–	–	+	–	+	+	+	11	20	0.252
6	–	+	–	+	+	–	+	+	+	–	+	+	–	–	–	+	+	+	–	+	12	20	0.132
7	–	–	–	–	+	–	+	–	–	+	–	+	–	+	–	–	–	+	+	–	7	20	0.132
8	–	–	–	–	+	+	+	+	+	–	–	–	–	+	+	–	+	–	+	+	10	20	0.412
9	–	–	–	–	–	–	–	+	+	+	+	+	+	–	+	+	+	+	+	+	12	20	0.132
10	–	–	–	–	+	+	+	+	–	+	+	–	–	+	–	+	+	–	+	+	11	20	0.252
11	–	–	+	+	+	+	–	–	+	+	+	+	–	–	–	+	–	–	–	–	9	20	0.412
12	–	–	+	+	–	+	+	–	+	+	+	+	–	+	+	–	–	+	–	+	12	20	0.132
13	–	–	–	+	–	–	–	–	–	+	+	+	–	–	+	–	+	–	+	+	8	20	0.252
14	–	+	–	+	–	+	–	–	+	+	+	+	+	+	+	–	+	+	+	+	14	20	0.021
15	–	–	–	–	+	+	+	+	+	+	+	–	–	+	+	+	–	+	+	+	13	20	0.058
16	–	+	–	–	–	+	+	–	+	+	–	+	–	–	+	+	+	+	+	–	11	20	0.252
17	–	–	–	–	–	–	+	+	–	+	–	–	–	–	–	+	+	+	+	–	7	20	0.132
18	–	–	–	–	+	–	+	+	+	–	+	+	–	+	–	+	+	+	+	+	12	20	0.132
19	–	–	–	+	+	+	–	+	+	+	+	–	–	–	–	–	+	+	–	+	10	20	0.412
20	–	–	–	+	+	–	+	+	+	–	–	–	–	–	–	+	+	+	+	+	10	20	0.412

Both sign test experiments tend to indicate the simulation is representative of the level of effort given by the Allied aircraft in the historical combat operations. In the case of Scenario 1 sortie hours, the bootstrap/sign test rejected the null hypothesis in 15% of the trials. With respect to Scenario 2 sortie hours, the bootstrap/sign test method rejected the null hypothesis in 5% of the trials. These results, in effect, bridge the gap between the previous validation methods, in which the simulation result for Scenario 1 sortie hours was statistically rejected and the result for Scenario 2 sortie hours was not rejected as statistically different. These conclusions provide a stronger indication of model acceptability than either of the previous tests for accepting the model as valid.

5.3.2 Scenario 1 MOEs

Previous analyses of Scenario 1 MOEs provided a somewhat mixed picture of the simulation's fidelity with respect to the historic data. The historic U-Boat kills total was slightly outside the simulation confidence interval, though the practical difference was negligible. Comparisons between the confidence interval generated by the historic monthly data and those generated from each iteration's monthly data, however, demonstrated 100% overlap, and hence, no statistical difference between the results from any individual iteration and the historic data. This approach also lacked any meaningful confidence when all such comparisons were taken together. The historic U-Boat sightings total was well within the confidence interval derived from the simulation data. The subsequent analysis with respect to the monthly means showed similar results to the U-Boat kills with the identical problem of providing no joint confidence.

Table 5.8 shows the bootstrap samples for Scenario 1 U-Boat sightings generated for comparison with the simulation results. The monthly bootstrap U-Boat sightings are totaled in the right-most column.

Table 5.8 Bootstrap U-Boat Sightings – Scenario 1

	U-Boat Sightings						
Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	14	18	10	42	42	42	168
2	18	14	42	18	19	18	129
3	18	18	19	18	19	14	106
4	10	14	14	14	42	14	108
5	14	19	42	32	42	19	168
6	42	18	32	32	42	14	180
7	19	32	14	32	18	19	134
8	18	14	14	10	14	42	112
9	18	19	18	42	18	19	134
10	32	32	32	32	18	18	164
11	32	10	19	14	10	32	117
12	10	19	42	32	10	32	145
13	32	19	19	42	18	18	148
14	32	32	42	42	42	10	200
15	10	32	14	18	18	32	124
16	32	32	10	18	42	14	148
17	19	19	14	19	19	32	122
18	32	19	42	18	32	14	157
19	10	19	19	32	32	32	144
20	32	42	10	32	42	14	172

Table 5.9 summarizes the sign test classifications for the paired data (X_i, Y_i) for Scenario 1 U-Boat sightings, where X_i is the i^{th} bootstrap U-Boat sightings total and Y_i is the U-Boat sightings total from the i^{th} simulation iteration from Table 4.7. The sign test statistic T and number of non-tied pairs n are displayed as well.

Table 5.9 Sign Test Calculations – U-Boat Sightings, Scenario 1

Observation	1	2	3	4	5	6	7	8	9	10
Sign	–	0	+	+	–	–	+	+	+	+
Observation	11	12	13	14	15	16	17	18	19	20
Sign	–	+	–	–	+	–	–	–	+	–
T	9									
n	19									

For n = 19, $P(t \leq 5) = 0.0358$ and $P(t \geq 13) = 0.0358$ defining an overall $(1 - \alpha) = 0.9284$. Since $5 < T = 9 < 13$, there is insufficient evidence to reject H_0. There is no compelling evidence to suggest the simulation does not faithfully represent the real-world system with respect to Scenario 1 U-Boat sightings. The resulting p-value is 0.5.

The results for the entire experiment are summarized in Table 5.10. Of the 20 sign test trials, the p-values ranged in value from 0.021 to 0.5. Under the rejection criteria, the null hypothesis was rejected in 3 of the 20 trials.

Table 5.10 Summary of 20 Bootstrap Experiments for Scenario 1 U-Boat Sightings

	Comparison Classification																						
Trail	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	T	n	p–value
1	–	0	+	+	–	–	+	+	+	+	–	+	–	–	+	–	–	–	+	–	9	19	0.500
2	–	0	+	+	–	–	+	+	+	+	–	+	–	–	+	–	–	–	+	–	9	19	0.500
3	–	–	–	+	–	–	–	+	+	+	–	+	–	+	+	–	+	+	+	–	10	20	0.412
4	–	–	+	–	–	–	–	+	+	+	–	+	–	–	–	–	–	+	+	–	7	20	0.132
5	0	–	–	+	–	–	+	–	+	+	–	+	–	–	+	–	+	+	+	–	9	19	0.500
6	+	–	+	+	–	+	+	–	+	+	–	–	–	–	–	–	–	0	+	–	8	19	0.324
7	–	–	–	+	–	+	–	–	+	+	–	+	–	–	+	–	–	+	+	+	9	20	0.412
8	–	+	+	+	+	+	+	–	+	+	–	+	+	–	–	–	+	+	+	+	14	20	0.021
9	–	–	+	–	+	+	+	–	+	–	–	+	–	0	–	–	–	–	+	+	8	19	0.324
10	–	+	–	+	+	+	+	–	+	+	+	+	+	–	–	+	–	+	+	+	14	20	0.021
11	–	–	+	–	+	–	–	+	+	+	–	+	–	+	–	+	+	+	+	–	11	20	0.252
12	+	+	–	–	+	+	+	–	–	+	–	+	–	–	–	–	–	+	+	–	9	20	0.412
13	+	–	+	+	–	+	–	+	+	+	–	–	–	–	+	–	+	–	+	+	11	20	0.252
14	–	–	+	+	+	+	+	+	+	+	–	+	–	–	–	–	–	+	–	–	10	20	0.412
15	–	–	–	+	–	–	–	–	+	+	–	+	–	–	+	+	–	+	–	+	8	20	0.252
16	–	–	–	+	+	+	–	+	+	+	–	+	–	–	+	–	–	+	+	–	10	20	0.412
17	+	–	–	+	+	+	+	+	+	+	–	+	+	+	–	–	0	–	+	–	12	19	0.084
18	–	–	–	–	–	–	+	–	+	+	–	+	–	–	–	–	–	+	–	–	5	20	0.021
19	–	+	+	+	–	+	–	+	+	+	–	+	–	+	–	+	–	–	+	+	12	20	0.132
20	–	–	+	+	+	+	–	–	–	+	+	+	–	+	+	–	–	+	+	+	12	20	0.132

Table 5.11 shows the bootstrap samples of Scenario 1 U-Boat kills generated for a single replication of the bootstrap/sign test experiment. The monthly bootstrap U-Boat kills are totaled in the right-most column.

Table 5.11 Bootstrap U-Boat Kills – Scenario 1

	U-Boat Kills						
Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	0	1	0	1	1	3
2	1	1	1	1	0	1	5
3	1	0	0	1	0	0	2
4	1	0	1	1	0	1	4
5	0	1	1	1	0	0	3
6	0	0	1	1	0	1	3
7	0	1	1	1	0	1	4
8	0	1	1	0	1	1	4
9	1	1	1	1	1	1	6
10	0	1	0	1	0	0	2
11	0	0	1	1	1	1	4
12	1	0	1	1	1	1	5
13	0	0	0	1	1	1	3
14	0	1	0	1	1	1	4
15	1	0	1	1	0	0	3
16	0	0	1	0	0	1	2
17	1	1	0	1	1	1	5
18	0	1	1	1	0	0	3
19	1	0	1	0	0	1	3
20	0	0	1	1	0	1	3

Table 5.12 summarizes the sign test classifications for the paired data (X_i, Y_i) for Scenario 1 U-Boat kills, where X_i is the i^{th} bootstrap U-Boat kills total and Y_i is the U-Boat kills total from the i^{th} simulation iteration from Table 4.8. The sign test statistic T and number of non-tied pairs n are displayed as well.

Table 5.12 Sign Test Calculations – U-Boat Kills, Scenario 1

Observation	1	2	3	4	5	6	7	8	9	10
Sign	–	0	+	–	+	–	+	–	–	+
Observation	11	12	13	14	15	16	17	18	19	20
Sign	0	–	–	–	+	+	–	+	+	+
T	9									
n	18									

For n = 18, P($t \le 5$) = 0.0481 and P($t \ge 12$) = 0.0481 defining an overall $(1 - \alpha)$ = 0.9038. Since $5 < T = 9 < 12$, there is insufficient evidence to reject H_0. There is no compelling evidence to suggest the simulation does not faithfully represent the real-world system with respect to Scenario 1 U-Boat kills. The resulting p-value is 0.4073.

The results for the entire experiment are summarized in Table 5.13. Of the 20 sign test trials, the p-values ranged in value from 0.011 to 0.5. Under the rejection criteria, the null hypothesis was rejected in 5 of the 20 trials.

Table 5.13 Summary of 20 Bootstrap Experiments for Scenario 1 U-Boat Kills

	Comparison Classification																						
Trial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	T	n	p–value
1	–	0	+	–	+	–	+	–	–	+	0	–	–	–	+	+	–	+	+	+	9	18	0.407
2	–	+	0	–	0	0	+	–	+	+	+	–	–	–	+	0	+	+	+	0	9	15	0.151
3	0	+	0	–	+	–	+	0	+	+	0	+	–	0	+	+	–	0	+	+	10	14	0.029
4	–	+	0	–	+	–	+	–	+	+	0	–	–	0	0	+	+	+	0	+	9	15	0.151
5	–	+	–	–	+	–	+	–	+	+	+	+	–	–	+	–	+	+	+	+	12	20	0.132
6	–	+	+	–	+	–	+	–	+	+	0	+	–	–	+	+	0	+	+	+	12	18	0.048
7	0	+	0	–	–	–	+	0	+	+	0	+	–	–	+	+	+	0	+	0	9	14	0.090
8	–	0	–	+	0	–	+	+	–	+	+	+	–	–	+	+	–	+	+	+	11	18	0.119
9	–	+	–	–	0	–	+	–	+	+	+	0	–	–	+	+	0	+	0	+	9	16	0.227
10	–	+	+	0	0	–	–	0	+	+	+	–	–	–	–	+	+	+	+	+	10	17	0.166
11	–	+	0	0	+	–	+	–	+	+	+	–	–	–	+	+	+	0	+	–	10	17	0.166
12	–	+	0	+	0	–	+	+	0	+	+	+	0	–	+	+	–	+	+	+	12	16	0.011
13	–	0	0	0	0	–	+	–	+	+	0	–	0	–	+	+	+	+	0	0	7	12	0.194
14	0	+	+	+	+	–	+	–	+	+	+	–	0	–	–	+	0	+	+	+	12	17	0.025
15	–	–	–	+	0	–	+	+	+	+	+	–	–	–	+	–	–	+	+	–	9	19	0.500
16	–	+	0	–	0	–	+	+	+	+	–	0	–	+	+	0	+	–	+	0	9	15	0.151
17	–	+	–	0	+	0	+	0	0	+	–	+	–	0	–	+	–	0	+	+	8	14	0.212
18	–	+	0	–	0	–	+	–	+	+	0	0	–	–	+	+	–	0	+	0	7	14	0.395
19	–	+	+	0	+	–	+	0	+	+	+	+	–	–	+	+	–	+	+	0	12	17	0.025
20	–	0	–	–	0	+	+	0	+	+	+	–	+	–	+	+	–	+	+	0	10	16	0.105

Both sign test experiments tend to indicate that the simulation is representative of historical combat operations for Scenario 1. In the case of Scenario 1 U-Boat sightings, the bootstrap/sign test rejected the null hypothesis in 15% of the trials. With respect to Scenario 1 U-Boat kills, the bootstrap/sign test method rejected the null hypothesis in 25% of the trials. Rather than make a validation conclusion based on a single statistical pass/fail, as in the first analysis method, the bootstrap/sign test methodology provides a broader context to the simulation results. These results, in effect, give broader insight into the validity of the simulation when the variability of the real-world operation is considered through the bootstrap. These conclusions provide stronger rationale than either of the previous tests for accepting the model as valid with respect to the MOEs.

5.3.3 Scenario 2 MOEs

Previous analyses of Scenario 2 MOEs provided a somewhat mixed picture of the simulation's fidelity with respect to the historic data. The historic U-Boat sightings total was slightly outside the simulation confidence interval, though the practical difference was negligible. Comparisons between the confidence interval generated by the historic monthly data and those generated from each iteration's monthly data, however, demonstrated 100% overlap, and hence, no statistical difference between the results from any individual iteration and the historic data. This approach, however, also lacked any meaningful confidence when all such comparisons were taken together. The historic U-Boat kills total was well within the confidence interval derived from the simulation data. The subsequent analysis with respect to the monthly means showed similar results to the sightings with the identical joint confidence problem.

Table 5.14 shows the bootstrap samples for Scenario 2 U-Boat sightings generated for a single replication of the bootstrap/sign test experiment. The monthly bootstrap U-Boat sightings are totaled in the right-most column.

Table 5.14 Bootstrap U-Boat Sightings – Scenario 2

	U-Boat Sightings						
Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	81	7	52	60	98	52	350
2	98	98	21	98	81	98	494
3	98	81	81	21	60	7	348
4	98	7	52	52	60	52	321
5	81	52	52	52	60	60	357
6	81	81	98	52	7	52	371
7	60	98	98	21	7	21	305
8	7	52	98	81	21	98	357
9	52	52	52	52	21	98	327
10	60	98	60	52	81	60	411
11	81	81	21	21	52	98	354
12	98	60	21	52	52	21	304
13	60	7	81	52	21	52	273
14	7	52	60	52	21	52	244
15	52	81	98	21	81	81	414
16	7	81	21	60	81	52	302
17	98	52	7	21	21	21	220
18	60	98	98	21	7	60	344
19	52	60	21	81	81	98	393
20	7	81	98	21	81	21	309

Table 5.15 summarizes the sign test classifications for the paired data (X_i, Y_i) for Scenario 2 U-Boat sightings, where X_i is the i^{th} bootstrap U-Boat sightings total and Y_i is the U-Boat sightings total from the i^{th} simulation iteration from Table 4.10. The sign test statistic T and number of non-tied pairs n are displayed as well.

Table 5.15 Sign Test Calculations – U-Boat Sightings, Scenario 2

Observation	1	2	3	4	5	6	7	8	9	10
Sign	–	–	–	–	–	–	+	–	+	–
Observation	11	12	13	14	15	16	17	18	19	20
Sign	–	+	+	+	–	+	+	+	–	+

T	9
n	20

For n = 20, P($t \leq 5$) = 0.0207 and P($t \geq 14$) = 0.0207 defining an overall $(1 - \alpha)$ = 0.9586. Since $5 < T = 9 < 14$, there is insufficient evidence to reject H_0. There is no compelling evidence to suggest the simulation does not faithfully represent the real-world system with respect to Scenario 2 U-Boat sightings. The resulting p-value is 0.4119.

The results for the entire experiment are summarized in Table 5.16. Of the 20 sign test trials, the p-values ranged in value from 0.058 to 0.412. Under the rejection criteria, the null hypothesis was not rejected in any of the 20 trials.

Table 5.16 Summary of 20 Bootstrap Experiments for Scenario 2 U-Boat Sightings

	Comparison Classification																						
Trial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	T	n	p–value
1	–	–	–	–	–	–	+	–	+	–	–	+	+	+	–	+	+	+	–	+	9	20	0.412
2	+	–	–	–	+	–	+	+	–	–	+	–	+	+	–	–	+	–	+	–	9	20	0.412
3	–	–	–	+	+	+	+	–	–	–	–	+	–	–	–	+	+	+	+	+	10	20	0.412
4	–	+	–	+	+	+	+	+	+	+	–	+	+	–	–	+	–	+	–	–	12	20	0.132
5	–	+	+	+	–	–	+	+	+	+	–	–	–	–	+	–	+	+	+	–	11	20	0.252
6	+	–	–	+	–	+	+	–	+	+	–	+	–	–	–	+	+	+	+	+	12	20	0.132
7	–	+	+	+	+	+	–	–	+	+	+	–	–	–	–	+	–	+	–	–	10	20	0.412
8	+	+	–	–	+	+	+	+	+	+	+	–	–	–	–	–	–	–	+	+	11	20	0.252
9	–	–	+	+	+	+	+	+	–	+	+	–	–	+	+	+	–	+	–	+	13	20	0.058
10	–	+	–	+	–	+	+	+	+	+	+	+	–	–	+	–	–	–	+	+	12	20	0.132
11	+	+	+	–	+	+	–	+	+	+	+	+	–	–	–	–	+	–	–	–	11	20	0.252
12	+	+	+	+	+	–	+	+	+	–	–	+	–	–	–	–	+	+	–	+	12	20	0.132
13	–	–	+	–	–	0	–	–	–	+	–	+	+	–	+	+	+	+	+	+	10	19	0.324
14	–	–	–	+	–	+	+	–	+	+	–	+	+	+	+	+	+	–	+	–	12	20	0.132
15	–	–	+	+	–	–	–	+	+	+	–	–	+	–	–	–	–	+	+	+	9	20	0.412
16	–	–	–	–	+	+	+	–	–	+	–	+	–	–	–	+	–	+	–	–	7	20	0.132
17	+	+	–	+	–	–	–	–	0	+	+	+	+	–	–	+	+	+	–	–	10	19	0.324
18	–	+	–	+	–	+	+	+	–	+	–	–	+	–	+	+	–	+	+	–	11	20	0.252
19	–	–	+	–	+	–	–	+	+	+	+	–	+	–	+	+	–	+	–	–	10	20	0.412
20	–	+	+	–	+	–	+	+	–	–	+	–	+	–	+	0	+	+	–	+	11	19	0.180

Table 5.17 shows the bootstrap samples of Scenario 2 U-Boat kills generated for a single replication of the bootstrap/sign test experiment. The monthly bootstrap U-Boat kills are totaled in the right-most column.

Table 5.17 Bootstrap U-Boat Kills – Scenario 2

	U-Boat Kills						
Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4	4	1	2	1	13	25
2	4	13	1	13	5	2	38
3	4	4	1	5	7	2	23
4	1	2	7	5	2	13	30
5	2	7	1	1	4	1	16
6	7	1	5	1	2	5	21
7	2	4	1	5	1	13	26
8	1	5	1	5	7	4	23
9	13	5	5	7	5	7	42
10	13	13	5	1	5	5	42
11	4	1	1	2	1	2	11
12	1	7	1	1	1	2	13
13	13	5	13	1	2	1	35
14	13	4	2	5	2	1	27
15	2	7	13	4	13	13	52
16	4	1	5	13	13	1	37
17	13	2	13	13	1	1	43
18	4	7	13	5	1	7	37
19	4	4	5	7	2	7	29
20	5	7	7	7	7	13	46

Table 5.18 summarizes the sign test classifications for the paired data (X_i, Y_i) for Scenario 2 U-Boat kills, where X_i is the i^{th} bootstrap U-Boat kills total and Y_i is the U-Boat kills total from the i^{th} simulation iteration from Table 4.11. The sign test statistic T and number of non-tied pairs n are displayed as well.

Table 5.18 Sign Test Calculations – U-Boat Kills, Scenario 2

Observation	1	2	3	4	5	6	7	8	9	10
Sign	+	–	+	0	+	+	+	+	–	–
Observation	11	12	13	14	15	16	17	18	19	20
Sign	+	+	–	0	–	0	–	–	+	–
T	9									
n	17									

For n = 17, $P(t \leq 4) = 0.0245$ and $P(t \geq 12) = 0.0245$ defining an overall $(1 - \alpha) = 0.9510$. Since $4 < T = 9 < 12$, there is insufficient evidence to reject H_0. There is no compelling evidence to suggest the simulation does not faithfully represent the real-world system with respect to Scenario 2 U-Boat kills. The resulting p-value is 0.3145.

The results for the entire experiment are summarized in Table 5.19. Of the 20 sign test trials, the p-values ranged in value from 0.058 to 0.5. Under the rejection criteria, the null hypothesis was not rejected in any of the 20 trials.

Table 5.19 Summary of 20 Bootstrap Experiments for Scenario 2 U-Boat Kills

	Comparison Classification																						
Trial	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	T	n	p–value
1	+	–	+	0	+	+	+	+	–	–	+	+	–	0	–	0	–	–	+	–	9	17	0.315
2	–	+	–	+	–	–	+	+	–	+	–	+	–	+	–	–	–	+	0	0	8	18	0.407
3	–	–	–	–	–	–	–	+	–	+	–	+	+	–	+	+	–	–	–	–	6	20	0.058
4	–	–	+	0	+	–	+	+	–	–	+	0	+	+	+	+	–	–	–	–	9	18	0.407
5	–	–	–	–	+	+	–	+	+	+	+	–	–	–	–	+	–	–	–	–	7	20	0.132
6	–	–	–	–	–	+	–	+	0	+	+	+	–	–	–	+	–	–	–	0	6	18	0.119
7	–	–	–	0	+	+	+	+	–	+	+	+	–	0	–	+	–	+	+	–	10	18	0.240
8	–	–	+	+	–	+	+	+	+	–	+	–	–	–	–	+	+	+	–	–	10	20	0.412
9	–	+	–	0	–	+	+	+	–	+	–	+	–	+	–	+	+	–	+	+	11	19	0.180
10	–	–	+	+	–	–	+	+	–	+	–	+	+	–	–	+	–	+	+	–	10	20	0.412
11	–	–	–	–	–	+	+	+	–	–	–	+	–	0	+	–	–	+	–	+	7	19	0.180
12	–	–	+	–	–	–	+	+	–	+	+	–	–	–	–	+	–	–	+	+	8	20	0.252
13	–	–	–	+	+	+	+	+	–	+	–	–	–	–	–	+	–	+	+	–	9	20	0.412
14	+	–	–	–	+	+	+	–	–	0	+	–	0	+	0	+	0	–	+	–	8	16	0.402
15	–	0	0	–	–	+	+	+	–	+	–	+	–	–	+	+	+	–	–	+	9	18	0.407
16	+	–	–	–	–	+	+	+	–	–	+	+	–	+	+	+	–	+	–	–	10	20	0.412
17	–	–	+	–	+	+	+	0	–	–	+	+	–	–	+	+	–	–	–	+	9	19	0.500
18	–	+	–	–	–	+	+	+	–	+	–	+	+	–	–	+	–	+	+	–	10	20	0.412
19	–	–	–	+	–	–	+	+	–	+	–	+	+	+	–	+	–	+	–	+	10	20	0.412
20	+	–	0	+	–	–	+	–	+	+	+	+	+	–	+	+	–	–	–	–	10	19	0.324

Both sign test experiments indicate the simulation is representative of historical combat operations for Scenario 2, since the null hypothesis was not rejected in 20 trials for either MOE. Though the original validation test showed a statistical difference in the number of U-Boat sightings, the results of the sign test may indicate the simulation was a better model than the single original test indicated. The monthly mean test demonstrated 100% overlap between the historic and simulation confidence intervals, though lacking in overall confidence. The conclusions drawn from the bootstrap/sign test methodology provide stronger indication than either of the previous tests for accepting the model as valid with respect to the MOEs.

5.3.4 Validation Conclusions

In the first validation analysis, a traditional statistical analysis was made between the overall MOE totals of the simulation and the real-world operations. These results varied by MOE. Using the traditional t-test, validation analysis provided a single pass/fail determination for each MOE. Half of the six tests made showed statistical difference between the simulation and historic data, although the practical differences were essentially negligible. Though the validation determination was favorable, the test assumed the historic outcome represented the mean of all such outcomes – a possibly risky assumption.

In the second validation analysis, an attempt to gain insight into the simulation's performance relative to the stochastic nature of the real-world process was made. The simulation appeared to perform exceedingly well against the real-world data in each experiment. However, due to the joint confidence dilemma discussed previously, little insight could be made with practical statistical confidence.

The proposed bootstrap/sign test validation methodology provides more information than the single pass/fail t-test of the first method and more statistical confidence than the confidence interval comparison of the second method. The sortie hour tests produced null hypothesis rejection rate of 15% for Scenario 1 and 5% for Scenario 2. The remaining MOEs for Scenario 1 produced a null hypothesis rejection rate of 15% for U-Boat sightings and 25% for U-Boat kills. Scenario 2 produced a null hypothesis rejection rate of 0% for both MOEs.

Ultimately, the validation determination rests with the decision maker, who takes risk, practical differences, and other associated costs into account. As an interesting example for demonstrating validation techniques, the model is sufficiently valid, and its success as an experimental platform has been demonstrated and well documented in [Champagne, *et al*, 2003], [Champagne and Hill, 2003], [Champagne, 2003], [Carl, 2003], [Carl, *et al*, 2003], and [Hill, *et al*, 2003a].

5.4 Contributions

The proposed bootstrap/sign test methodology goes beyond the traditional model validation methods. Using the historic data as a single sample from the distribution underlying the real-world system, bootstrap samples were generated and tested against the simulation data using the sign test. Multiple replications were made to give an indication of how well the simulation data compared to the bootstrap data sets by providing more than a single pass/fail. Instead, the multiple replications provide a rate of pass/fail that does not suffer the same analytical dilemma found in the second method demonstrated. These tests, therefore, provide a stronger indication of the extent to which the simulation data represents the real-world system than the traditional MOE validation using the t-test.

VI. Contributions and Avenues for Future Research

This research was not intended to advocate agent-based modeling. Rather, this research objectively investigates agent-based models for combat simulation applications. This research had two major objectives with respect to agent-based combat modeling. The first was to demonstrate the applicability of the agent-based paradigm on the modeling of real-world combat scenarios. This involved the creation of an agent-based combat model that conformed to the concepts of agent-based systems found in the vast majority of the literature and validated against a substantial real-world combat operation. The second objective was to develop a framework through which the validation of agent-based combat scenarios could be tested. This chapter summarizes the research, highlights the original contributions, and identifies possible avenues for further research. A detailed discussion of data and results accompanies the presentation of methodologies and analyses in Chapter IV and Chapter V.

6.1 Contributions

Chapter I defined four principal research areas in support of the objectives. The contributions made by this research are presented in the context of these areas.

6.1.1 Establishing the Background and Supporting Work

The state-of-the-art in agent-based combat simulation was established through a comprehensive review of the literature. The literature review identified complementary agent-based modeling in the fields of AI, artificial life, and heuristic optimization. Additionally, the literature review established that the majority of agent-based research

diverged from combat modeling by concentrating on cooperative agents. Through identification of the strengths inherent in the foundational fields, a link was established between agent-based combat simulation and human behavior modeling.

Agent-based combat modeling is in its infancy, and while the literature suggests agent-based methods hold promise to gain insights into the effects of human behavior on the outcome of combat, deficiencies exist in both the agent-based approach and in the scope of combat operations addressed. More work is needed to establish the viability of agent-based models for combat analysis.

6.1.2 Extend Agent-Based Combat Simulations to the Mission-Level

Within the context of this research effort, an agent-based combat simulation of the Allied offensive search for U-Boats in the Bay of Biscay during WW II was researched, defined, and built. The simulation models continuous combat over (2 distinct) six months of operations. This presents two demonstrable contributions. First, agent-based simulations were extended to the mission-level for the first time. Second, agent-based simulations were shown applicable to real-world combat scenarios.

An additional contribution demonstrated in the building of the Bay of Biscay agent-based simulation is the development of a methodology whereby historical combat is encapsulated into an agent-based model. The development process in Chapter IV stressed several areas necessary for establishing the credibility of agent-based combat simulation results, particularly: 1) determining and parameterizing the underlying agent behaviors; 2) researching the model parameterizations required for historical accuracy;

and 3) quantifying the sufficiency of the model emergent behavior with respect to the historical record.

Finally, the offensive search scenario was decomposed to provide a methodology for extending the Bay of Biscay scenario to other, possibly more relevant, scenarios. These applications are quite varied and encompassed military, law enforcement, treaty verification, and homeland security.

6.1.3 Develop Validation Methods for Agent-Based Combat Simulations

Several contributions were made in the area of agent-based model verification and validation. Prior to this effort, the V&V literature lacked a taxonomy that included agent-based methods. This research developed a V&V taxonomy based on technique functionality and included agent-based simulation validation methods.

In showing the veracity of the Bay of Biscay agent-based simulation, additional contributions are made to simulation V&V. Primarily, output analysis techniques were extended to incorporate the validation of the emergent behavior of the agents.

Finally, a novel statistical validation methodology was developed to determine model veracity with respect to the stochastic process underlying the real-world combat operations. The technique combines two nonparametric techniques, the bootstrapping and sign test, to provide more information than was available through more traditional methods such as the t-test.

6.1.4 Demonstration of Methods via Known Use-Case

Several practical contributions were made through the presentation of this research. First, a well-accepted modeling and simulation process was demonstrated in the development of the Bay of Biscay model. Second, the use of techniques classified in the V&V taxonomy, presented in Chapter III, was demonstrated in Chapter IV to establish several levels of validation for the simulation.

6.2 Future Research

The contributions of this research effort immediately suggest several promising areas for follow-on research. Some of these are outlined below.

6.2.1 Additional Agent Behaviors

In building the Bay of Biscay agent-based simulation, the emphasis was on showing applicability against the real-world historic outcome. Agent behavior was not addressed beyond reproducing known behavior as documented in the historic accounts. Thus far, then, the behavioral aspects of agent-based simulation have not been explored.

Future research would extend this research into behavioral realms. For example, information-based decision making could be explored via routing choices, submergence policy, search zone selection, and search pattern type. These decisions could factor in both time and location for various contact types (sighting, attacks, and kills). Additionally, behavioral focused agent-based combat simulation could provide additional avenues into the development of tactics, doctrine, or policy.

Though some aspects of adaptation were explored in [Price, 2003; Hill, *et al*, 2003a], adaptive agent routing and search is ripe for exploration.

6.2.2 Modern Scenario Extensions

Chapter IV presented a methodology for extending the modeled offensive search scenario to applications that are more relevant to modern concerns. There is a great opportunity to explore these extensions through the model development and V&V approaches demonstrated in Chapter IV.

Appendix A. Bootstrap Results for Simulation MOEs

The following tables contain the bootstrap samples produced for the analysis in Chapter V. The MOEs are presented in the same order as the analyses within the body of this text.

A.1 Scenario 1 Sortie Hours

Table A.1 Bootstrap Samples, Replication 1, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,400	3,400	4,100	4,600	4,100	3,130	22,730
2	3,400	3,130	4,100	3,130	4,400	4,600	22,760
3	3,400	3,400	4,100	4,400	4,600	4,600	24,500
4	4,100	4,600	4,600	4,400	4,600	3,130	25,430
5	4,600	4,600	4,100	4,100	3,400	4,100	24,900
6	4,600	4,100	4,400	4,100	4,600	4,400	26,200
7	4,600	3,400	4,100	4,400	4,100	3,130	23,730
8	4,100	4,400	4,100	4,400	4,600	3,400	25,000
9	3,130	4,400	4,400	4,600	4,400	4,600	25,530
10	3,130	3,130	4,400	4,100	4,400	4,400	23,560
11	3,400	4,100	4,100	4,600	4,100	4,600	24,900
12	4,100	4,600	4,100	4,100	4,100	4,600	25,600
13	3,130	4,400	3,130	4,100	4,600	4,100	23,460
14	3,130	3,400	4,600	4,400	4,600	4,100	24,230
15	4,600	4,600	3,130	3,400	3,130	3,130	21,990
16	3,400	4,100	4,400	3,130	3,130	4,100	22,260
17	3,130	4,600	3,130	3,130	4,100	4,100	22,190
18	3,400	4,600	3,130	4,400	4,100	4,600	24,230
19	3,400	3,400	4,400	4,600	4,600	4,600	25,000
20	4,400	4,600	4,600	4,600	4,600	3,130	25,930

Table A.2 Bootstrap Samples, Replication 2, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,130	4,100	4,600	4,600	3,130	4,100	23,660
2	4,400	4,100	4,400	4,400	4,400	3,130	24,830
3	4,600	4,600	4,600	4,600	4,400	4,100	26,900
4	3,130	3,400	3,130	3,130	3,130	3,400	19,320
5	4,600	3,130	3,130	4,600	4,600	4,100	24,160
6	4,400	4,600	4,100	4,400	3,130	4,100	24,730
7	4,100	3,400	4,100	4,100	3,130	4,400	23,230
8	4,400	3,130	3,130	3,130	4,600	4,400	22,790
9	4,100	3,130	3,400	4,600	4,600	3,130	22,960
10	3,130	3,400	4,600	4,600	4,600	4,600	24,930
11	4,400	3,130	4,600	4,400	3,400	4,600	24,530
12	3,130	4,600	4,600	4,100	4,600	3,130	24,160
13	3,400	4,400	4,100	3,130	3,400	4,600	23,030
14	3,130	4,400	3,130	4,400	4,100	4,100	23,260
15	3,400	4,400	4,600	3,130	4,400	4,600	24,530
16	3,130	4,600	4,400	3,130	4,600	3,130	22,990
17	3,130	4,400	4,100	3,130	4,600	3,130	22,490
18	4,600	3,130	4,600	4,600	4,100	3,400	24,430
19	4,100	4,100	3,400	4,600	4,400	4,600	25,200
20	4,400	3,130	3,130	4,100	3,130	4,600	22,490

Table A.3 Bootstrap Samples, Replication 3, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,600	4,600	3,130	3,130	4,100	4,600	24,160
2	3,400	4,100	3,130	4,600	4,400	4,100	23,730
3	4,100	4,400	3,130	4,600	4,600	4,600	25,430
4	3,400	4,400	4,600	3,400	3,130	4,600	23,530
5	4,400	4,600	4,600	4,400	4,600	4,400	27,000
6	4,100	4,400	4,600	4,400	4,600	4,600	26,700
7	4,600	4,600	4,400	3,130	4,400	3,400	24,530
8	3,400	4,600	3,400	4,600	4,600	4,600	25,200
9	3,130	4,100	4,400	4,400	3,130	4,600	23,760
10	4,600	4,100	4,100	3,400	3,400	3,130	22,730
11	4,400	3,130	4,600	4,600	4,100	3,130	23,960
12	4,100	4,100	4,100	4,600	4,600	4,600	26,100
13	4,100	4,600	4,600	3,130	4,400	4,100	24,930
14	4,400	4,600	4,600	4,600	4,400	3,400	26,000
15	3,130	4,100	4,600	4,100	4,600	4,600	25,130
16	4,600	4,600	4,400	4,100	4,100	4,600	26,400
17	4,400	4,400	4,400	4,600	4,100	4,600	26,500
18	3,130	4,600	3,130	4,600	3,130	3,400	21,990
19	4,100	4,600	4,600	4,400	3,130	3,400	24,230
20	3,400	4,100	4,100	4,600	4,100	4,400	24,700

Table A.4 Bootstrap Samples, Replication 4, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,600	4,600	4,100	4,400	4,600	3,400	25,700
2	4,400	4,400	4,600	4,600	4,400	4,400	26,800
3	3,130	4,600	3,400	3,130	3,130	3,400	20,790
4	4,600	4,600	3,130	3,400	3,130	3,400	22,260
5	3,400	4,100	4,600	4,100	3,400	3,130	22,730
6	3,130	4,400	3,400	4,100	3,400	4,400	22,830
7	4,400	4,400	4,600	4,600	3,400	4,600	26,000
8	4,100	4,400	3,400	3,400	4,600	4,600	24,500
9	3,400	4,100	4,100	4,400	4,100	3,130	23,230
10	4,400	4,600	3,130	4,600	4,100	4,600	25,430
11	3,400	4,600	4,400	4,600	4,100	3,130	24,230
12	4,400	4,600	4,100	4,400	3,130	4,400	25,030
13	4,600	3,400	3,130	4,100	4,100	4,400	23,730
14	4,600	3,400	4,100	4,600	3,400	4,400	24,500
15	4,600	4,600	4,600	4,100	3,130	4,600	25,630
16	4,600	4,100	4,100	4,600	3,130	4,600	25,130
17	4,100	4,600	4,100	4,400	4,400	4,600	26,200
18	3,130	4,100	4,400	4,400	3,400	4,600	24,030
19	3,130	3,400	3,400	4,600	4,600	4,600	23,730
20	4,400	4,400	4,600	3,400	3,130	4,600	24,530

Table A.5 Bootstrap Samples, Replication 5, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,400	4,400	4,100	4,400	4,400	3,130	23,830
2	3,130	4,100	3,400	4,100	4,600	4,600	23,930
3	4,600	3,400	3,130	4,400	4,100	4,100	23,730
4	4,100	3,400	3,400	4,100	4,600	3,400	23,000
5	4,400	3,400	4,600	3,130	4,600	4,600	24,730
6	4,400	4,400	4,600	4,600	3,130	4,100	25,230
7	3,400	4,600	4,600	3,400	4,100	4,100	24,200
8	3,130	3,130	3,130	3,400	4,600	4,600	21,990
9	4,100	4,600	3,130	4,400	3,130	4,600	23,960
10	4,400	4,400	4,600	3,130	4,100	4,100	24,730
11	3,400	3,400	4,400	4,600	4,400	3,400	23,600
12	4,400	3,130	4,100	4,400	4,400	4,600	25,030
13	4,600	4,600	4,600	3,130	4,100	4,600	25,630
14	3,400	4,600	4,100	4,600	4,400	4,600	25,700
15	4,600	3,400	4,400	4,600	4,400	4,600	26,000
16	3,130	3,130	4,600	3,400	4,100	3,400	21,760
17	4,600	4,100	4,100	4,600	4,600	4,600	26,600
18	4,600	4,100	4,600	3,400	3,400	4,100	24,200
19	4,400	4,600	4,600	4,600	4,100	4,600	26,900
20	4,600	4,600	4,600	4,600	4,400	4,600	27,400

Table A.6 Bootstrap Samples, Replication 6, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,600	3,400	4,100	4,100	3,130	4,600	23,930
2	3,400	4,100	3,400	3,400	3,400	4,600	22,300
3	4,100	3,400	3,130	4,400	4,600	4,100	23,730
4	4,600	4,100	4,600	4,600	4,600	3,400	25,900
5	3,130	4,600	4,600	4,600	3,400	4,600	24,930
6	3,400	4,600	4,400	4,600	3,130	4,600	24,730
7	4,600	3,130	4,100	4,600	4,100	4,100	24,630
8	3,130	4,100	4,600	4,400	4,600	4,400	25,230
9	4,600	4,600	4,600	4,600	3,400	4,100	25,900
10	4,600	4,600	3,400	4,600	3,130	4,400	24,730
11	3,400	4,100	3,400	4,400	4,600	3,130	23,030
12	4,600	4,600	4,600	3,130	4,100	4,100	25,130
13	4,400	4,100	4,100	4,600	4,600	4,100	25,900
14	4,600	3,400	4,600	4,100	4,400	4,100	25,200
15	4,100	4,600	3,400	4,400	4,100	3,400	24,000
16	4,400	4,600	4,600	4,400	4,600	4,600	27,200
17	4,100	4,400	3,130	4,100	4,100	3,400	23,230
18	3,400	4,100	4,600	4,600	3,400	4,100	24,200
19	4,100	4,600	4,400	4,600	3,130	3,400	24,230
20	4,400	4,600	4,100	4,600	4,100	3,400	25,200

Table A.7 Bootstrap Samples, Replication 7, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,400	3,130	3,130	4,400	4,600	3,400	23,060
2	4,600	4,400	4,100	3,400	3,400	3,130	23,030
3	4,400	3,400	4,400	3,400	4,100	4,400	24,100
4	4,600	4,600	3,130	3,130	3,400	3,400	22,260
5	4,600	4,600	4,600	4,100	4,600	4,600	27,100
6	3,400	4,400	4,400	4,600	4,600	4,600	26,000
7	3,400	4,600	4,100	4,600	4,100	4,400	25,200
8	4,600	4,400	3,400	4,600	4,600	4,100	25,700
9	3,130	3,400	4,100	3,130	4,400	4,400	22,560
10	4,600	4,400	3,130	4,100	4,100	4,400	24,730
11	4,400	3,400	4,100	4,100	4,600	4,600	25,200
12	4,100	4,400	4,400	4,600	4,100	4,400	26,000
13	4,600	4,400	4,600	4,400	3,130	4,100	25,230
14	4,400	3,130	3,400	4,600	4,600	4,600	24,730
15	4,600	4,600	4,600	3,400	4,400	4,100	25,700
16	4,600	4,400	4,600	3,400	3,130	3,400	23,530
17	4,600	4,600	4,100	3,400	4,600	3,400	24,700
18	4,600	4,600	4,100	4,600	4,600	4,100	26,600
19	3,130	3,130	3,130	4,100	4,600	4,600	22,690
20	4,600	4,100	3,130	4,400	3,400	3,400	23,030

Table A.8 Bootstrap Samples, Replication 8, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,600	4,600	3,400	4,600	4,600	4,100	25,900
2	4,600	4,600	4,600	4,600	3,400	3,130	24,930
3	3,400	3,130	4,600	4,600	3,400	4,100	23,230
4	3,130	4,100	4,600	4,400	4,600	3,130	23,960
5	4,600	4,600	4,600	4,600	3,130	4,100	25,630
6	3,400	4,400	4,600	3,400	4,400	3,400	23,600
7	4,400	3,400	4,100	4,400	4,400	4,600	25,300
8	4,600	3,130	4,100	3,130	4,100	4,600	23,660
9	3,400	4,400	3,130	3,400	4,100	4,600	23,030
10	3,130	3,400	4,100	4,400	3,400	4,600	23,030
11	3,130	4,600	4,600	4,400	3,130	4,400	24,260
12	4,600	4,100	4,600	4,400	4,600	4,100	26,400
13	4,600	3,130	3,400	3,400	4,100	4,100	22,730
14	4,600	3,400	4,100	4,100	4,400	4,400	25,000
15	4,600	4,600	3,130	4,600	4,100	4,600	25,630
16	3,400	4,400	4,400	3,130	4,100	3,400	22,830
17	4,100	3,400	4,600	3,400	4,100	4,100	23,700
18	4,400	3,130	4,100	4,600	4,400	4,400	25,030
19	4,600	4,400	4,100	3,400	4,400	3,400	24,300
20	4,600	3,400	4,400	4,600	4,600	4,400	26,000

Table A.9 Bootstrap Samples, Replication 9, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,400	4,100	4,600	3,130	4,600	4,400	25,230
2	3,130	3,400	3,130	3,400	3,400	4,400	20,860
3	4,600	4,100	4,600	4,600	4,600	4,600	27,100
4	3,130	4,100	3,400	4,100	4,600	4,100	23,430
5	4,400	4,100	4,600	4,600	4,600	3,400	25,700
6	4,600	3,130	4,400	4,600	4,600	4,100	25,430
7	3,400	4,600	4,600	3,400	4,600	4,100	24,700
8	3,400	4,400	4,100	4,600	3,400	4,400	24,300
9	3,130	4,100	3,400	4,600	4,600	3,400	23,230
10	4,400	4,400	4,600	3,130	3,400	4,400	24,330
11	4,600	4,400	4,600	3,400	4,100	3,400	24,500
12	3,130	3,130	3,130	3,400	4,600	3,400	20,790
13	4,600	4,600	4,600	3,130	4,100	4,600	25,630
14	4,600	4,100	4,600	4,100	4,400	4,600	26,400
15	3,130	4,100	3,400	3,130	3,130	4,100	20,990
16	3,130	4,100	4,600	4,600	4,400	4,100	24,930
17	3,400	3,130	4,400	4,600	4,100	3,130	22,760
18	4,100	4,600	4,100	4,600	4,400	4,600	26,400
19	4,100	4,100	4,600	4,600	4,100	4,400	25,900
20	4,100	3,400	4,400	3,400	4,600	4,100	24,000

Table A.10 Bootstrap Samples, Replication 10, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,400	4,400	3,130	4,100	4,100	4,600	23,730
2	4,600	4,100	4,400	4,400	4,400	4,100	26,000
3	4,600	3,130	3,400	3,400	3,400	4,600	22,530
4	4,600	3,130	4,100	3,130	4,600	4,600	24,160
5	3,130	4,400	3,400	4,400	4,100	3,400	22,830
6	4,400	4,600	4,400	4,600	3,400	3,400	24,800
7	3,400	3,400	4,600	3,130	4,100	4,100	22,730
8	3,130	4,600	4,600	3,400	4,600	4,100	24,430
9	3,130	4,600	4,600	4,600	3,130	4,100	24,160
10	4,600	4,400	3,130	4,100	3,130	3,400	22,760
11	4,600	4,600	4,600	4,400	3,400	4,100	25,700
12	4,600	4,600	4,100	3,130	4,100	4,100	24,630
13	4,100	3,130	4,100	4,400	4,400	4,100	24,230
14	4,400	3,400	3,400	3,130	4,600	4,100	23,030
15	4,600	4,600	4,100	4,600	3,130	4,100	25,130
16	3,400	4,600	4,600	4,400	4,600	3,130	24,730
17	3,130	4,400	4,600	3,130	4,600	3,400	23,260
18	4,600	3,400	4,100	4,400	3,130	4,600	24,230
19	4,600	3,130	3,130	4,400	3,400	3,130	21,790
20	4,600	4,600	4,600	3,400	4,100	3,400	24,700

Table A.11 Bootstrap Samples, Replication 11, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,130	3,400	4,100	4,600	3,400	4,600	23,230
2	4,600	4,100	4,600	4,100	4,600	4,600	26,600
3	4,400	3,130	3,400	4,600	4,400	4,400	24,330
4	4,400	4,400	4,100	3,130	3,130	4,600	23,760
5	4,400	4,100	4,600	3,130	4,600	3,400	24,230
6	4,400	3,400	4,100	3,400	4,100	4,100	23,500
7	4,600	4,600	4,600	4,600	4,400	3,130	25,930
8	4,600	4,100	3,400	4,400	4,600	3,130	24,230
9	3,400	4,400	3,400	4,600	4,400	4,600	24,800
10	4,600	3,400	3,130	4,400	3,400	4,600	23,530
11	3,130	3,400	3,400	4,600	4,600	3,400	22,530
12	4,600	3,130	4,600	3,130	3,400	4,400	23,260
13	4,100	4,100	4,600	3,130	4,100	3,400	23,430
14	4,600	4,400	4,400	3,130	4,400	4,400	25,330
15	4,600	4,600	4,400	3,400	3,400	3,400	23,800
16	3,130	4,400	3,400	4,600	3,400	3,130	22,060
17	3,130	4,600	3,400	3,130	4,600	3,130	21,990
18	3,130	3,400	4,600	3,130	3,130	4,600	21,990
19	4,600	4,100	4,600	4,400	3,400	3,400	24,500
20	4,100	4,600	4,100	4,600	4,100	3,130	24,630

Table A.12 Bootstrap Samples, Replication 12, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,130	4,600	3,400	4,600	4,600	4,600	24,930
2	4,100	3,400	4,600	3,130	4,100	4,600	23,930
3	4,400	3,130	3,400	4,400	3,400	4,100	22,830
4	3,130	4,600	4,600	4,100	3,400	3,400	23,230
5	4,100	4,600	4,600	4,100	4,400	3,400	25,200
6	3,130	4,600	3,400	3,130	4,400	3,400	22,060
7	4,400	4,400	3,130	4,600	4,400	4,600	25,530
8	4,400	4,600	4,400	4,600	4,600	4,400	27,000
9	4,600	3,400	4,400	4,600	4,600	3,400	25,000
10	4,100	4,600	4,600	4,600	4,100	4,600	26,600
11	4,400	4,600	3,130	4,400	3,400	4,100	24,030
12	4,100	4,400	4,600	3,400	4,100	3,130	23,730
13	3,400	3,130	3,400	4,100	4,600	4,100	22,730
14	3,130	3,400	4,600	4,100	4,100	4,600	23,930
15	4,600	3,130	4,100	3,400	4,400	3,400	23,030
16	4,600	3,400	4,400	3,400	4,400	4,100	24,300
17	4,600	4,400	4,600	3,400	4,100	4,600	25,700
18	4,100	4,600	4,600	3,130	4,100	4,600	25,130
19	3,130	4,100	4,100	3,400	4,600	4,400	23,730
20	3,400	4,400	3,130	4,600	4,400	3,130	23,060

Table A.13 Bootstrap Samples, Replication 13, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,600	3,130	3,130	4,600	4,600	4,600	24,660
2	4,400	4,600	3,130	3,130	4,100	3,400	22,760
3	4,400	3,400	4,100	3,400	3,400	3,130	21,830
4	4,600	4,600	3,130	4,100	4,600	4,400	25,430
5	4,600	4,400	4,600	4,400	4,400	4,400	26,800
6	4,600	3,130	3,130	4,600	4,600	3,130	23,190
7	4,600	4,400	4,400	4,600	3,130	3,400	24,530
8	4,400	4,600	4,400	3,130	4,600	4,100	25,230
9	4,600	3,400	4,400	4,400	4,400	4,600	25,800
10	3,400	4,600	4,600	4,600	4,400	4,600	26,200
11	4,600	3,130	4,600	4,400	3,400	4,600	24,730
12	4,400	3,400	4,600	4,400	4,600	4,100	25,500
13	4,100	4,400	3,400	4,100	4,100	3,130	23,230
14	4,100	3,130	3,400	3,130	3,400	4,100	21,260
15	4,400	4,100	4,100	3,400	4,600	4,600	25,200
16	4,100	3,130	4,600	4,400	4,400	4,600	25,230
17	3,130	4,400	4,600	4,400	4,100	4,600	25,230
18	4,600	4,100	4,600	4,600	3,130	4,400	25,430
19	4,100	4,400	4,600	3,130	4,100	3,130	23,460
20	3,130	3,130	3,400	4,600	4,100	3,400	21,760

Table A.14 Bootstrap Samples, Replication 14, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,130	4,400	4,100	4,600	4,100	4,400	24,730
2	4,100	4,600	4,100	4,400	4,600	4,400	26,200
3	4,400	4,100	4,600	3,400	4,600	4,400	25,500
4	3,130	4,600	4,400	4,100	4,400	3,400	24,030
5	4,600	4,400	4,600	4,100	4,400	4,100	26,200
6	3,400	4,600	4,400	4,400	3,130	3,400	23,330
7	3,130	3,400	4,400	4,100	3,130	4,400	22,560
8	4,600	3,130	4,100	4,400	3,400	4,400	24,030
9	3,130	4,100	3,400	4,400	3,400	4,400	22,830
10	4,400	4,100	4,100	3,400	4,100	4,600	24,700
11	3,400	3,400	4,600	3,400	4,600	3,130	22,530
12	3,130	4,600	4,600	4,100	3,400	4,100	23,930
13	4,600	4,100	3,130	4,600	4,400	4,600	25,430
14	4,600	4,600	3,400	3,130	4,600	4,600	24,930
15	4,600	3,130	3,400	3,400	4,100	4,600	23,230
16	3,400	4,600	3,400	4,600	3,130	4,600	23,730
17	3,400	3,130	4,400	3,400	3,400	4,100	21,830
18	4,600	3,130	4,600	3,130	4,600	4,600	24,660
19	4,600	4,100	3,400	4,600	3,130	4,100	23,930
20	3,400	4,600	4,600	3,400	4,600	3,400	24,000

Table A.15 Bootstrap Samples, Replication 15, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,130	4,600	4,100	4,600	3,130	4,600	24,160
2	4,100	4,600	4,400	4,600	4,400	3,400	25,500
3	3,400	4,400	3,130	4,400	4,600	4,600	24,530
4	3,130	4,600	3,400	4,400	4,100	4,400	24,030
5	4,600	4,600	4,400	4,600	4,600	4,600	27,400
6	3,400	3,130	4,600	4,600	4,100	4,100	23,930
7	4,100	4,400	4,600	4,600	4,400	4,400	26,500
8	3,400	4,400	4,100	4,600	4,100	4,100	24,700
9	4,600	4,600	4,600	3,400	4,600	4,100	25,900
10	3,400	3,130	4,100	3,130	3,130	4,600	21,490
11	4,100	3,400	4,100	4,600	4,100	4,600	24,900
12	4,100	4,600	4,400	4,600	4,400	4,100	26,200
13	4,600	4,600	4,600	4,100	4,100	4,600	26,600
14	3,130	4,100	3,400	4,600	3,130	4,100	22,460
15	3,400	3,400	4,100	4,600	4,600	3,400	23,500
16	4,600	4,100	3,400	3,130	4,600	4,400	24,230
17	4,400	3,130	3,130	4,400	3,130	4,100	22,290
18	4,600	3,130	4,400	4,400	3,130	3,130	22,790
19	4,400	3,400	3,400	4,600	4,100	4,600	24,500
20	4,100	4,600	4,600	3,130	4,100	3,400	23,930

Table A.16 Bootstrap Samples, Replication 16, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,100	4,600	4,100	3,130	4,600	4,600	25,130
2	4,400	4,600	4,100	3,130	4,600	3,400	24,230
3	4,400	4,100	3,130	3,130	4,600	4,600	23,960
4	4,400	4,600	3,130	4,100	4,600	3,400	24,230
5	4,400	3,130	4,400	3,130	4,600	4,100	23,760
6	3,400	3,130	3,130	4,400	4,600	3,400	22,060
7	4,600	4,400	3,130	4,100	4,600	4,600	25,430
8	4,100	4,600	4,600	4,600	4,400	4,100	26,400
9	4,100	3,130	4,600	4,400	4,400	3,400	24,030
10	4,600	3,130	4,600	3,400	3,130	3,130	21,990
11	3,130	3,400	4,400	4,600	4,600	3,130	23,260
12	4,600	3,130	4,100	3,130	4,100	4,400	23,460
13	3,400	4,400	4,600	4,100	3,400	3,130	23,030
14	3,130	4,400	4,400	3,130	4,100	3,400	22,560
15	4,100	4,600	3,130	4,600	3,400	4,600	24,430
16	4,600	4,600	4,600	3,130	4,400	4,400	25,730
17	4,600	4,600	4,100	3,130	3,130	3,130	22,690
18	3,130	4,600	4,100	4,100	4,600	4,600	25,130
19	4,100	4,600	4,600	4,600	4,400	4,100	26,400
20	3,130	4,400	4,600	4,100	4,100	3,400	23,730

Table A.17 Bootstrap Samples, Replication 17, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,400	4,600	3,130	4,600	4,100	4,600	25,430
2	3,400	4,600	4,600	4,100	3,400	4,600	24,700
3	3,400	4,100	3,400	4,600	4,600	4,600	24,700
4	4,100	3,130	4,400	3,400	4,600	4,100	23,730
5	4,600	3,400	4,600	4,400	4,600	4,400	26,000
6	4,100	4,600	4,600	4,600	3,400	4,600	25,900
7	4,600	4,600	4,100	3,400	3,400	4,600	24,700
8	4,600	4,600	3,130	4,600	3,130	4,100	24,160
9	3,130	4,600	4,100	3,130	3,400	4,600	22,960
10	4,600	3,400	4,400	4,100	4,100	3,400	24,000
11	4,400	3,130	3,130	4,400	4,600	4,400	24,060
12	4,600	4,600	4,600	4,100	4,600	3,130	25,630
13	3,400	4,600	4,100	4,400	3,130	4,100	23,730
14	4,400	4,400	3,400	3,130	4,600	4,400	24,330
15	3,400	4,400	4,600	4,100	4,100	3,130	23,730
16	4,600	3,130	4,600	4,600	3,400	3,130	23,460
17	4,100	3,130	4,400	4,100	4,600	4,600	24,930
18	4,400	4,600	4,400	4,600	3,400	4,600	26,000
19	4,100	3,130	3,130	3,130	3,130	4,400	21,020
20	3,400	4,600	4,600	3,400	3,130	3,130	22,260

Table A.18 Bootstrap Samples, Replication 18, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,400	3,400	4,100	4,600	4,100	4,600	25,200
2	3,400	3,130	4,400	4,100	3,130	3,130	21,290
3	4,600	3,130	4,600	3,400	4,600	4,600	24,930
4	4,400	3,400	3,130	4,400	4,600	3,130	23,060
5	3,400	4,400	4,100	4,400	3,400	4,600	24,300
6	3,130	4,100	4,600	3,400	4,600	3,130	22,960
7	4,100	4,600	4,100	3,130	3,400	4,100	23,430
8	4,100	4,400	4,600	4,100	4,600	4,100	25,900
9	4,600	4,600	4,100	3,130	4,100	3,130	23,660
10	4,100	4,400	4,400	3,400	4,600	4,400	25,300
11	4,100	4,600	4,100	4,600	4,600	4,600	26,600
12	3,400	4,600	3,400	4,100	3,130	4,400	23,030
13	3,130	4,600	3,130	4,100	4,600	3,130	22,690
14	4,100	4,600	3,400	4,100	4,600	4,600	25,400
15	4,600	4,600	4,600	3,130	4,600	4,600	26,130
16	4,100	3,130	3,400	4,600	4,600	4,400	24,230
17	4,600	4,600	3,130	4,400	4,100	4,100	24,930
18	4,100	3,130	4,400	3,400	4,100	4,100	23,230
19	3,130	4,600	4,400	4,600	3,130	4,100	23,960
20	3,130	4,600	3,400	3,400	3,130	4,600	22,260

Table A.19 Bootstrap Samples, Replication 19, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,100	3,400	3,400	4,600	4,400	4,100	24,000
2	4,100	3,400	4,600	3,400	4,100	4,600	24,200
3	3,130	4,100	4,600	3,130	4,600	4,400	23,960
4	3,400	3,130	3,130	4,100	4,600	3,400	21,760
5	4,600	3,130	4,100	4,600	4,400	3,130	23,960
6	4,600	3,130	3,400	4,100	4,400	3,130	22,760
7	3,130	4,400	4,600	4,600	4,400	4,100	25,230
8	3,400	4,400	4,100	3,130	4,600	4,600	24,230
9	3,130	3,400	4,400	4,600	3,400	4,600	23,530
10	3,400	4,100	4,600	4,100	4,600	4,600	25,400
11	3,400	3,130	3,400	4,600	4,100	4,100	22,730
12	4,400	4,400	4,600	3,130	4,400	3,130	24,060
13	3,130	4,600	4,600	3,130	4,100	3,130	22,690
14	4,600	4,400	3,400	3,400	4,600	4,600	25,000
15	3,130	3,130	4,600	4,100	3,130	4,600	22,690
16	4,600	4,600	3,130	4,600	4,400	4,100	25,430
17	3,130	4,600	3,130	4,600	3,400	3,130	21,990
18	4,600	3,400	4,600	3,400	3,130	3,130	22,260
19	4,400	4,400	3,130	4,400	4,100	4,600	25,030
20	4,400	3,130	3,130	4,400	4,600	4,600	24,260

Table A.20 Bootstrap Samples, Replication 20, Scenario 1 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	3,130	4,600	3,400	3,400	4,400	3,130	22,060
2	3,130	4,600	4,600	4,600	4,600	3,400	24,930
3	3,400	3,400	4,400	3,400	4,600	4,100	23,300
4	4,600	3,130	4,600	4,600	4,400	3,130	24,460
5	4,600	3,400	4,100	4,600	4,400	4,100	25,200
6	4,400	3,130	3,400	3,400	4,100	4,600	23,030
7	3,400	3,400	4,600	4,600	4,400	3,400	23,800
8	3,130	4,100	4,600	3,400	4,400	4,600	24,230
9	4,600	3,400	3,400	4,100	4,600	4,400	24,500
10	3,130	3,400	3,400	3,400	4,400	4,100	21,830
11	4,100	4,600	4,100	4,600	4,400	4,600	26,400
12	3,130	4,400	4,600	4,600	4,100	3,130	23,960
13	4,100	4,600	4,100	4,600	3,130	3,400	23,930
14	4,600	4,400	4,600	4,600	3,400	4,600	26,200
15	4,600	3,130	3,400	4,400	3,400	3,130	22,060
16	4,600	3,400	4,400	3,400	4,100	4,400	24,300
17	3,130	4,400	4,600	4,400	4,600	4,100	25,230
18	4,600	3,130	4,400	4,600	4,600	4,600	25,930
19	4,100	3,400	4,100	3,130	3,400	3,130	21,260
20	4,400	4,600	3,400	4,400	4,400	4,600	25,800

A.2 Scenario 2 Sortie Hours

Table A.21 Bootstrap Samples, Replication 1, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5,900	5,350	7,000	5,900	7,000	7,000	38,150
2	8,000	8,700	8,700	7,000	5,900	7,000	45,300
3	5,900	5,900	4,200	4,200	7,000	8,700	35,900
4	7,000	8,000	5,900	4,200	8,000	8,700	41,800
5	8,000	4,200	8,700	5,900	5,350	4,200	36,350
6	7,000	5,900	7,000	8,000	4,200	5,900	38,000
7	7,000	7,000	7,000	7,000	4,200	7,000	39,200
8	5,350	5,350	8,700	5,350	5,900	5,350	36,000
9	4,200	5,350	7,000	8,700	5,350	5,350	35,950
10	7,000	8,000	7,000	8,700	8,700	7,000	46,400
11	8,000	5,350	8,700	7,000	8,700	5,350	43,100
12	5,350	8,700	5,900	8,000	4,200	7,000	39,150
13	8,700	8,000	5,350	8,000	5,900	4,200	40,150
14	4,200	8,700	5,350	7,000	5,900	5,900	37,050
15	8,700	8,000	5,350	5,900	4,200	8,700	40,850
16	8,700	5,350	7,000	8,700	5,900	5,350	41,000
17	8,700	5,900	4,200	5,350	8,700	8,000	40,850
18	4,200	4,200	5,350	8,700	8,700	8,700	39,850
19	5,900	7,000	7,000	5,350	8,700	5,350	39,300
20	4,200	7,000	8,000	8,700	5,350	4,200	37,450

Table A.22 Bootstrap Samples, Replication 2, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,200	8,700	8,700	5,350	5,900	7,000	39,850
2	5,900	4,200	5,900	4,200	7,000	5,900	33,100
3	5,350	5,900	5,900	8,700	5,900	8,000	39,750
4	8,700	5,900	8,000	8,700	4,200	5,900	41,400
5	7,000	5,350	8,700	5,350	5,900	7,000	39,300
6	8,700	8,700	7,000	5,900	5,900	5,350	41,550
7	5,350	4,200	5,350	8,000	5,350	7,000	35,250
8	8,700	8,700	8,000	4,200	4,200	4,200	38,000
9	7,000	8,700	4,200	4,200	5,900	5,900	35,900
10	5,900	7,000	4,200	8,700	5,900	5,900	37,600
11	5,900	4,200	5,350	5,900	5,900	5,350	32,600
12	7,000	8,700	5,900	5,350	5,350	8,700	41,000
13	8,000	8,000	8,000	4,200	8,000	5,350	41,550
14	8,000	7,000	5,350	5,900	5,350	5,900	37,500
15	8,700	5,900	8,700	8,700	5,900	5,900	43,800
16	5,350	4,200	8,000	5,900	7,000	5,350	35,800
17	8,700	4,200	4,200	4,200	5,350	5,350	32,000
18	5,350	5,900	7,000	8,000	5,350	4,200	35,800
19	8,700	4,200	5,900	4,200	4,200	5,350	32,550
20	5,900	7,000	5,900	5,900	5,350	8,700	38,750

Table A.23 Bootstrap Samples, Replication 3, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7,000	7,000	7,000	8,700	7,000	5,350	42,050
2	8,000	4,200	8,700	5,900	8,700	8,000	43,500
3	8,700	7,000	8,000	5,900	8,700	5,350	43,650
4	5,350	8,700	5,350	5,900	5,350	8,000	38,650
5	4,200	7,000	5,900	4,200	7,000	7,000	35,300
6	4,200	4,200	4,200	5,350	8,000	8,700	34,650
7	8,700	5,900	8,000	7,000	5,900	4,200	39,700
8	5,900	5,900	4,200	5,900	8,000	5,350	35,250
9	8,000	8,000	8,000	8,700	8,700	5,900	47,300
10	5,900	5,350	8,700	7,000	5,350	4,200	36,500
11	8,000	4,200	5,900	5,350	7,000	7,000	37,450
12	8,000	7,000	5,350	5,350	5,350	8,000	39,050
13	4,200	8,700	5,900	8,000	5,900	8,700	41,400
14	5,900	5,350	7,000	5,350	7,000	5,900	36,500
15	8,000	5,900	7,000	5,900	5,350	5,900	38,050
16	8,000	8,000	5,900	5,350	8,700	7,000	42,950
17	8,000	7,000	8,000	5,350	5,350	5,900	39,600
18	7,000	8,000	8,700	5,900	5,900	4,200	39,700
19	5,350	5,350	8,700	8,700	5,350	4,200	37,650
20	5,350	8,000	8,000	8,700	5,350	5,900	41,300

Table A.24 Bootstrap Samples, Replication 4, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,700	5,900	4,200	5,350	8,700	8,700	41,550
2	5,900	7,000	8,000	7,000	8,000	8,000	43,900
3	8,700	5,350	5,350	5,900	5,900	8,000	39,200
4	8,700	7,000	7,000	5,350	5,350	7,000	40,400
5	8,000	8,000	5,350	5,350	7,000	5,350	39,050
6	8,700	8,000	8,700	7,000	7,000	4,200	43,600
7	5,350	5,350	8,000	5,350	8,000	5,350	37,400
8	8,000	4,200	8,000	5,350	5,350	4,200	35,100
9	5,350	5,350	5,350	8,700	5,350	5,350	35,450
10	5,350	4,200	7,000	8,000	8,000	8,700	41,250
11	5,900	8,000	5,350	5,350	5,350	4,200	34,150
12	8,700	5,900	7,000	5,350	5,350	4,200	36,500
13	5,350	8,700	4,200	4,200	5,350	5,900	33,700
14	5,350	4,200	7,000	7,000	5,350	7,000	35,900
15	8,000	4,200	7,000	8,700	5,900	5,900	39,700
16	8,700	7,000	5,350	5,350	5,350	5,900	37,650
17	4,200	4,200	5,900	5,350	4,200	5,900	29,750
18	4,200	5,900	5,350	7,000	8,000	5,900	36,350
19	5,350	8,000	5,900	5,350	8,700	5,900	39,200
20	7,000	5,900	5,900	5,350	8,000	8,000	40,150

Table A.25 Bootstrap Samples, Replication 5, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,000	8,700	4,200	8,000	5,350	5,350	39,600
2	4,200	5,350	5,350	5,350	8,700	7,000	35,950
3	5,900	8,700	5,350	5,350	5,900	4,200	35,400
4	5,900	7,000	4,200	8,000	8,700	4,200	38,000
5	7,000	4,200	7,000	8,000	4,200	8,700	39,100
6	8,000	5,900	8,000	8,700	5,900	5,900	42,400
7	8,700	5,900	4,200	8,000	8,000	4,200	39,000
8	5,350	8,700	7,000	5,350	7,000	5,350	38,750
9	8,000	8,700	7,000	5,350	5,900	5,350	40,300
10	5,350	8,700	7,000	8,000	8,000	8,000	45,050
11	5,350	8,000	8,700	8,000	8,700	5,350	44,100
12	7,000	5,900	5,900	7,000	5,350	8,700	39,850
13	5,900	7,000	5,900	5,350	7,000	5,900	37,050
14	5,350	8,700	5,900	5,350	5,350	8,700	39,350
15	8,700	8,700	5,350	8,000	8,700	8,700	48,150
16	5,350	8,000	8,700	4,200	5,350	4,200	35,800
17	8,700	7,000	8,700	7,000	5,350	5,350	42,100
18	5,900	7,000	4,200	7,000	7,000	5,900	37,000
19	4,200	5,350	5,350	5,350	5,350	8,000	33,600
20	8,700	4,200	5,350	8,000	8,700	4,200	39,150

Table A.26 Bootstrap Samples, Replication 6, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,000	7,000	8,700	7,000	5,900	8,000	44,600
2	5,900	4,200	5,900	8,700	4,200	4,200	33,100
3	8,000	4,200	5,900	8,700	5,900	8,000	40,700
4	4,200	5,350	8,700	5,900	7,000	8,000	39,150
5	4,200	5,350	7,000	8,000	8,700	7,000	40,250
6	5,900	8,700	8,000	4,200	7,000	8,700	42,500
7	5,350	8,000	4,200	8,000	5,900	7,000	38,450
8	4,200	5,900	4,200	8,000	8,700	5,900	36,900
9	8,700	7,000	4,200	8,700	4,200	4,200	37,000
10	8,000	8,700	4,200	8,000	8,000	7,000	43,900
11	5,900	8,000	8,700	4,200	4,200	4,200	35,200
12	5,350	5,900	8,000	7,000	5,350	5,350	36,950
13	5,350	8,700	8,000	8,700	7,000	8,700	46,450
14	4,200	5,900	5,350	7,000	8,700	7,000	38,150
15	4,200	7,000	8,700	5,350	5,350	8,700	39,300
16	8,000	4,200	5,350	5,900	7,000	4,200	34,650
17	7,000	4,200	4,200	8,700	7,000	7,000	38,100
18	5,900	8,000	5,900	8,000	5,900	4,200	37,900
19	8,700	8,700	8,000	4,200	8,700	4,200	42,500
20	8,000	8,000	4,200	4,200	5,350	8,700	38,450

Table A.27 Bootstrap Samples, Replication 7, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,000	7,000	5,350	8,000	8,000	4,200	40,550
2	4,200	5,900	7,000	8,700	5,900	8,700	40,400
3	7,000	5,900	5,350	5,350	8,700	8,000	40,300
4	8,700	8,000	7,000	7,000	8,000	5,900	44,600
5	5,900	5,900	4,200	8,000	5,900	8,700	38,600
6	4,200	7,000	8,700	8,700	5,350	8,000	41,950
7	7,000	7,000	7,000	4,200	4,200	7,000	36,400
8	8,000	8,000	4,200	8,700	7,000	8,700	44,600
9	5,900	8,000	8,700	4,200	8,000	5,900	40,700
10	7,000	7,000	4,200	8,700	5,350	5,350	37,600
11	5,900	8,000	8,700	8,700	5,350	8,700	45,350
12	4,200	5,350	7,000	4,200	8,000	5,900	34,650
13	8,000	8,700	5,350	5,900	7,000	4,200	39,150
14	5,350	4,200	5,900	5,900	8,000	7,000	36,350
15	5,900	8,000	5,900	8,700	4,200	8,000	40,700
16	8,000	8,000	7,000	5,350	5,900	5,900	40,150
17	5,900	8,000	8,700	5,350	5,900	7,000	40,850
18	5,350	8,700	8,000	5,900	7,000	4,200	39,150
19	5,900	5,900	5,350	7,000	8,700	5,350	38,200
20	5,900	5,350	8,700	7,000	5,900	8,700	41,550

Table A.28 Bootstrap Samples, Replication 8, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,000	5,350	8,000	7,000	8,700	4,200	41,250
2	5,350	7,000	7,000	7,000	8,700	5,900	40,950
3	8,700	4,200	5,350	4,200	8,700	8,000	39,150
4	7,000	7,000	8,000	7,000	5,900	8,000	42,900
5	5,350	7,000	5,350	5,350	4,200	8,000	35,250
6	5,900	5,900	8,700	7,000	4,200	8,000	39,700
7	7,000	5,350	8,700	5,350	4,200	8,000	38,600
8	5,900	5,350	4,200	8,700	5,350	5,900	35,400
9	8,000	4,200	8,000	4,200	7,000	4,200	35,600
10	5,350	8,000	8,700	7,000	8,700	8,000	45,750
11	8,700	8,700	5,350	5,350	5,900	7,000	41,000
12	7,000	8,000	7,000	5,350	8,000	8,700	44,050
13	5,350	5,350	5,900	5,900	8,700	8,700	39,900
14	4,200	8,700	4,200	7,000	4,200	8,000	36,300
15	5,350	4,200	8,000	5,350	4,200	4,200	31,300
16	5,350	8,700	5,900	8,000	5,350	8,000	41,300
17	4,200	8,700	8,700	4,200	4,200	8,700	38,700
18	8,000	8,700	8,000	8,000	8,700	8,000	49,400
19	4,200	5,350	8,000	5,900	5,350	5,900	34,700
20	5,900	5,350	7,000	5,350	4,200	7,000	34,800

Table A.29 Bootstrap Samples, Replication 9, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7,000	8,700	5,350	8,000	5,900	5,900	40,850
2	8,700	8,000	5,900	5,350	5,900	8,000	41,850
3	5,350	8,700	8,700	5,350	8,000	5,900	42,000
4	8,000	8,700	5,900	7,000	5,350	8,000	42,950
5	7,000	7,000	7,000	7,000	8,700	8,700	45,400
6	8,700	5,900	5,900	8,000	5,900	5,900	40,300
7	8,000	5,900	8,700	8,700	5,350	7,000	43,650
8	8,700	5,900	5,350	7,000	5,900	8,000	40,850
9	5,350	5,900	7,000	7,000	5,350	4,200	34,800
10	5,900	5,900	8,700	5,350	7,000	4,200	37,050
11	5,350	5,350	8,000	4,200	8,000	7,000	37,900
12	8,700	8,700	4,200	5,900	7,000	5,900	40,400
13	4,200	5,350	5,350	8,700	5,900	4,200	33,700
14	8,000	8,000	7,000	5,900	8,700	8,000	45,600
15	4,200	4,200	7,000	5,900	5,900	5,350	32,550
16	5,350	5,350	4,200	8,000	5,350	8,000	36,250
17	4,200	5,900	5,900	7,000	4,200	8,000	35,200
18	8,000	5,350	8,000	5,900	5,900	5,350	38,500
19	8,700	7,000	5,350	8,000	5,900	4,200	39,150
20	8,700	5,350	5,350	8,000	4,200	5,350	36,950

Table A.30 Bootstrap Samples, Replication 10, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,200	4,200	4,200	8,700	8,700	8,700	38,700
2	8,000	5,900	4,200	5,350	8,000	8,000	39,450
3	8,000	7,000	5,350	4,200	5,900	8,000	38,450
4	7,000	7,000	8,700	7,000	5,350	5,350	40,400
5	4,200	5,900	7,000	7,000	5,350	8,000	37,450
6	5,900	8,000	5,900	4,200	5,900	5,900	35,800
7	5,900	8,000	5,900	4,200	7,000	8,000	39,000
8	8,700	4,200	5,900	4,200	8,000	8,700	39,700
9	8,700	7,000	7,000	8,700	4,200	5,350	40,950
10	7,000	4,200	8,000	8,000	8,700	4,200	40,100
11	7,000	4,200	4,200	8,000	5,900	4,200	33,500
12	5,900	4,200	8,000	8,700	8,700	8,700	44,200
13	4,200	5,350	8,000	8,000	5,350	5,900	36,800
14	4,200	5,900	5,900	4,200	8,000	5,350	33,550
15	8,000	7,000	5,350	4,200	7,000	8,700	40,250
16	8,000	5,350	4,200	8,700	5,900	7,000	39,150
17	7,000	5,900	5,350	5,900	7,000	4,200	35,350
18	7,000	8,700	4,200	8,700	7,000	8,000	43,600
19	5,350	4,200	5,350	8,700	8,700	8,700	41,000
20	5,350	8,700	4,200	4,200	7,000	8,000	37,450

Table A.31 Bootstrap Samples, Replication 11, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7,000	7,000	7,000	5,900	4,200	5,350	36,450
2	7,000	5,350	7,000	7,000	5,350	4,200	35,900
3	5,900	7,000	4,200	4,200	4,200	7,000	32,500
4	5,350	7,000	5,900	4,200	8,000	8,000	38,450
5	5,350	8,700	8,700	5,350	4,200	8,000	40,300
6	7,000	5,350	4,200	4,200	5,900	7,000	33,650
7	5,350	5,350	8,700	8,000	8,000	8,000	43,400
8	8,700	4,200	7,000	4,200	8,700	8,700	41,500
9	8,000	8,000	4,200	5,350	5,900	8,700	40,150
10	5,900	7,000	5,350	5,900	8,700	5,900	38,750
11	5,350	4,200	8,000	4,200	7,000	8,700	37,450
12	8,700	7,000	5,900	4,200	8,000	4,200	38,000
13	7,000	8,000	8,000	5,900	5,900	5,900	40,700
14	8,000	7,000	8,000	8,000	8,000	5,900	44,900
15	5,900	8,700	8,000	4,200	7,000	5,350	39,150
16	7,000	5,350	4,200	5,900	7,000	8,700	38,150
17	8,000	5,350	8,000	4,200	8,000	8,700	42,250
18	4,200	8,700	8,000	4,200	8,700	8,700	42,500
19	7,000	8,700	8,700	8,700	4,200	5,350	42,650
20	8,700	8,700	4,200	8,000	8,000	8,000	45,600

Table A.32 Bootstrap Samples, Replication 12, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,700	5,350	4,200	4,200	8,000	8,000	38,450
2	7,000	8,000	8,000	5,900	7,000	8,700	44,600
3	7,000	5,350	5,350	4,200	8,700	4,200	34,800
4	5,900	7,000	8,000	4,200	8,000	5,350	38,450
5	7,000	7,000	7,000	8,000	7,000	5,900	41,900
6	5,900	5,900	4,200	4,200	4,200	8,700	33,100
7	8,700	5,900	4,200	7,000	7,000	4,200	37,000
8	8,700	4,200	8,700	8,700	8,700	7,000	46,000
9	7,000	5,900	4,200	7,000	5,900	5,900	35,900
10	4,200	8,000	4,200	8,000	8,700	4,200	37,300
11	4,200	7,000	7,000	8,700	5,350	4,200	36,450
12	8,000	4,200	4,200	5,900	7,000	5,350	34,650
13	7,000	5,900	8,000	8,700	5,350	4,200	39,150
14	7,000	5,900	8,000	5,900	4,200	4,200	35,200
15	4,200	8,000	7,000	4,200	5,350	4,200	32,950
16	5,350	8,700	8,700	8,700	7,000	5,900	44,350
17	5,350	8,000	8,700	7,000	5,900	7,000	41,950
18	8,700	8,700	5,350	8,000	4,200	5,350	40,300
19	8,700	8,000	5,350	7,000	7,000	5,350	41,400
20	5,900	7,000	7,000	8,000	7,000	5,350	40,250

Table A.33 Bootstrap Samples, Replication 13, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7,000	8,000	7,000	5,900	7,000	5,900	40,800
2	8,000	8,000	5,350	4,200	4,200	5,350	35,100
3	8,000	5,900	4,200	5,900	8,700	8,000	40,700
4	4,200	7,000	5,350	5,900	8,700	7,000	38,150
5	8,000	5,350	5,350	8,000	5,900	8,700	41,300
6	5,900	5,350	8,000	8,700	8,700	5,900	42,550
7	5,350	5,900	8,700	5,350	8,700	8,700	42,700
8	8,000	8,000	5,900	5,900	5,900	8,000	41,700
9	8,000	7,000	8,700	5,900	5,350	7,000	41,950
10	8,000	5,900	5,350	5,900	5,350	8,700	39,200
11	7,000	4,200	8,000	7,000	8,000	4,200	38,400
12	5,350	8,700	7,000	5,900	5,350	5,350	37,650
13	5,350	8,700	7,000	8,000	5,350	8,700	43,100
14	8,000	5,350	7,000	5,900	5,900	8,000	40,150
15	8,700	8,700	4,200	4,200	5,350	7,000	38,150
16	8,000	5,350	8,000	5,350	5,900	8,000	40,600
17	4,200	8,700	5,350	5,900	7,000	4,200	35,350
18	8,700	8,000	8,700	8,000	5,900	7,000	46,300
19	5,900	8,700	5,350	7,000	5,900	4,200	37,050
20	7,000	5,350	8,000	5,900	7,000	4,200	37,450

Table A.34 Bootstrap Samples, Replication 14, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,000	4,200	7,000	5,350	5,900	5,350	35,800
2	8,700	4,200	4,200	5,350	5,350	4,200	32,000
3	5,350	4,200	5,900	8,700	8,000	5,900	38,050
4	4,200	8,000	5,900	8,000	5,350	7,000	38,450
5	8,700	8,700	8,000	5,350	8,700	7,000	46,450
6	5,900	4,200	4,200	5,900	8,700	8,700	37,600
7	7,000	8,700	8,000	5,900	5,350	7,000	41,950
8	5,350	8,700	8,700	8,000	7,000	8,700	46,450
9	7,000	8,000	5,900	5,900	5,350	5,350	37,500
10	4,200	5,900	5,900	7,000	7,000	5,900	35,900
11	5,350	5,350	5,900	8,700	8,700	4,200	38,200
12	7,000	7,000	8,700	5,900	7,000	5,350	40,950
13	4,200	8,000	8,000	5,350	4,200	4,200	33,950
14	4,200	5,900	5,350	4,200	4,200	8,000	31,850
15	5,350	8,000	5,900	4,200	8,700	5,900	38,050
16	5,350	8,700	5,350	7,000	7,000	8,000	41,400
17	4,200	8,000	5,900	8,000	5,350	7,000	38,450
18	5,350	7,000	4,200	7,000	5,350	5,350	34,250
19	4,200	4,200	4,200	8,000	7,000	5,350	32,950
20	4,200	5,900	7,000	5,350	7,000	4,200	33,650

Table A.35 Bootstrap Samples, Replication 15, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5,900	5,350	5,900	7,000	5,900	8,000	38,050
2	4,200	8,700	7,000	8,700	5,900	5,350	39,850
3	5,900	7,000	5,350	5,350	8,000	5,900	37,500
4	8,000	5,900	8,700	7,000	8,700	7,000	45,300
5	8,000	5,350	8,000	5,900	4,200	7,000	38,450
6	5,350	5,900	7,000	7,000	4,200	4,200	33,650
7	4,200	7,000	5,350	4,200	5,900	5,350	32,000
8	7,000	8,000	5,900	8,700	5,900	4,200	39,700
9	8,000	8,000	5,900	4,200	5,350	7,000	38,450
10	8,000	4,200	4,200	4,200	5,350	4,200	30,150
11	8,700	4,200	5,350	5,900	4,200	8,700	37,050
12	7,000	8,700	8,000	5,350	7,000	8,700	44,750
13	4,200	5,350	8,700	5,900	5,900	8,700	38,750
14	5,900	4,200	7,000	5,350	7,000	7,000	36,450
15	5,900	7,000	7,000	4,200	8,700	5,900	38,700
16	7,000	4,200	7,000	7,000	5,350	5,900	36,450
17	8,000	8,000	8,000	5,900	8,700	4,200	42,800
18	5,350	7,000	7,000	4,200	5,900	5,900	35,350
19	8,700	8,000	4,200	7,000	4,200	8,000	40,100
20	4,200	7,000	8,700	4,200	7,000	4,200	35,300

Table A.36 Bootstrap Samples, Replication 16, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5,350	4,200	7,000	4,200	8,000	5,350	34,100
2	5,900	5,900	4,200	4,200	8,700	5,350	34,250
3	8,000	8,000	5,350	8,000	5,900	8,700	43,950
4	5,900	7,000	8,000	8,000	8,000	5,900	42,800
5	7,000	5,900	8,000	7,000	8,000	5,350	41,250
6	8,000	5,900	5,900	5,900	7,000	7,000	39,700
7	8,700	8,000	5,900	8,700	4,200	5,350	40,850
8	8,700	8,700	5,900	8,700	4,200	5,900	42,100
9	4,200	8,000	5,350	4,200	8,000	8,700	38,450
10	5,900	8,000	5,350	5,900	5,350	7,000	37,500
11	5,350	8,700	7,000	8,000	7,000	5,350	41,400
12	5,900	5,350	4,200	5,350	5,350	5,900	32,050
13	7,000	7,000	8,000	8,000	4,200	4,200	38,400
14	5,900	7,000	4,200	4,200	8,700	7,000	37,000
15	7,000	8,000	4,200	5,900	5,900	5,350	36,350
16	4,200	8,700	7,000	4,200	8,000	4,200	36,300
17	8,000	7,000	7,000	7,000	5,900	4,200	39,100
18	4,200	7,000	7,000	5,350	7,000	8,700	39,250
19	5,900	5,900	8,700	5,900	5,900	8,000	40,300
20	4,200	8,000	8,700	7,000	8,700	8,700	45,300

Table A.37 Bootstrap Samples, Replication 17, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,200	8,000	7,000	8,700	5,900	4,200	38,000
2	5,350	5,350	5,900	5,350	5,350	8,700	36,000
3	5,900	7,000	5,350	7,000	8,000	8,000	41,250
4	8,700	8,700	8,700	5,900	7,000	5,900	44,900
5	5,350	5,350	8,700	8,000	8,000	8,000	43,400
6	8,000	8,000	7,000	5,350	8,000	8,000	44,350
7	5,900	5,900	5,350	5,350	8,000	4,200	34,700
8	5,350	4,200	5,900	8,000	8,000	8,000	39,450
9	5,350	5,900	5,350	8,700	8,000	8,700	42,000
10	5,900	5,350	7,000	8,700	5,900	5,350	38,200
11	8,000	8,000	8,000	8,000	8,700	8,000	48,700
12	8,700	4,200	8,700	4,200	8,000	8,700	42,500
13	5,350	5,350	8,000	4,200	5,900	8,000	36,800
14	8,000	7,000	5,350	8,700	7,000	7,000	43,050
15	8,000	8,000	8,700	5,900	4,200	8,700	43,500
16	4,200	8,000	4,200	7,000	5,350	8,000	36,750
17	5,350	7,000	7,000	7,000	5,350	8,000	39,700
18	5,900	5,900	5,350	8,000	4,200	4,200	33,550
19	8,700	5,350	7,000	4,200	8,000	4,200	37,450
20	8,700	5,350	8,000	5,900	5,900	8,000	41,850

Table A.38 Bootstrap Samples, Replication 18, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,700	7,000	5,350	5,900	8,000	5,900	40,850
2	7,000	5,900	8,700	5,350	4,200	8,700	39,850
3	8,700	8,700	8,700	8,700	5,900	8,000	48,700
4	8,700	5,350	4,200	8,000	7,000	7,000	40,250
5	4,200	5,900	4,200	5,350	5,350	4,200	29,200
6	5,350	8,700	8,700	8,700	4,200	8,000	43,650
7	7,000	8,000	5,350	4,200	7,000	7,000	38,550
8	5,900	8,700	8,000	8,000	4,200	5,900	40,700
9	7,000	4,200	7,000	8,700	4,200	4,200	35,300
10	4,200	8,700	8,000	8,000	5,900	8,700	43,500
11	7,000	7,000	4,200	5,350	5,900	4,200	33,650
12	4,200	8,700	4,200	8,000	4,200	7,000	36,300
13	5,900	8,000	8,700	7,000	5,350	4,200	39,150
14	4,200	5,350	8,000	5,350	7,000	5,900	35,800
15	4,200	8,700	7,000	8,000	5,350	8,000	41,250
16	7,000	7,000	5,900	4,200	8,700	5,900	38,700
17	5,350	7,000	8,700	7,000	5,350	5,350	38,750
18	4,200	8,700	7,000	4,200	8,000	5,900	38,000
19	8,000	4,200	8,000	8,000	8,000	4,200	40,400
20	7,000	4,200	8,000	7,000	5,900	4,200	36,300

Table A.39 Bootstrap Samples, Replication 19, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4,200	8,000	5,900	8,700	5,350	4,200	36,350
2	5,350	5,350	8,700	8,700	7,000	4,200	39,300
3	4,200	8,700	8,700	7,000	7,000	4,200	39,800
4	5,900	4,200	5,350	8,700	5,350	8,700	38,200
5	7,000	4,200	8,700	5,350	5,350	8,000	38,600
6	7,000	8,700	5,900	8,000	5,900	4,200	39,700
7	5,900	7,000	8,700	5,900	8,000	5,900	41,400
8	7,000	8,000	8,000	5,350	5,900	5,350	39,600
9	5,900	5,900	8,700	4,200	7,000	8,700	40,400
10	5,350	7,000	4,200	7,000	7,000	5,900	36,450
11	5,350	5,900	8,000	8,000	5,350	4,200	36,800
12	8,700	5,900	7,000	8,000	8,700	8,700	47,000
13	8,700	8,700	4,200	5,350	7,000	8,000	41,950
14	5,350	8,000	7,000	8,000	7,000	5,900	41,250
15	5,900	8,700	8,000	7,000	4,200	5,900	39,700
16	7,000	7,000	8,700	5,350	8,000	5,350	41,400
17	8,000	4,200	5,350	7,000	7,000	7,000	38,550
18	4,200	5,900	8,000	8,700	5,900	8,000	40,700
19	8,700	7,000	8,700	4,200	8,000	7,000	43,600
20	5,350	8,700	8,000	4,200	8,000	5,900	40,150

Table A.40 Bootstrap Samples, Replication 20, Scenario 2 Sortie Hours

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	8,700	5,350	8,700	4,200	4,200	8,700	39,850
2	8,700	4,200	5,350	5,350	4,200	8,700	36,500
3	5,900	8,000	8,000	4,200	7,000	4,200	37,300
4	7,000	4,200	7,000	4,200	8,700	7,000	38,100
5	5,350	5,900	4,200	8,000	5,900	8,000	37,350
6	4,200	8,700	8,700	7,000	5,350	8,700	42,650
7	7,000	5,900	7,000	8,000	8,700	4,200	40,800
8	8,000	8,700	7,000	4,200	5,900	7,000	40,800
9	5,900	4,200	8,700	8,000	5,900	4,200	36,900
10	5,350	7,000	8,700	7,000	8,000	5,900	41,950
11	7,000	8,000	5,900	8,000	5,900	7,000	41,800
12	8,700	7,000	8,700	5,350	8,700	8,700	47,150
13	8,000	5,900	4,200	5,900	5,350	7,000	36,350
14	7,000	7,000	5,900	8,000	5,900	4,200	38,000
15	8,700	8,000	7,000	5,900	5,350	5,900	40,850
16	5,900	8,700	4,200	4,200	5,350	4,200	32,550
17	8,000	5,900	8,000	4,200	5,900	5,900	37,900
18	5,900	4,200	7,000	7,000	7,000	4,200	35,300
19	4,200	8,700	4,200	5,900	8,700	8,000	39,700
20	5,350	4,200	8,000	5,900	8,000	7,000	38,450

A.3 Scenario 1 U-Boat Sightings

Table A.41 Bootstrap Samples, Replication 1, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	14	18	10	42	42	42	168
2	18	14	42	18	19	18	129
3	18	18	19	18	19	14	106
4	10	14	14	14	42	14	108
5	14	19	42	32	42	19	168
6	42	18	32	32	42	14	180
7	19	32	14	32	18	19	134
8	18	14	14	10	14	42	112
9	18	19	18	42	18	19	134
10	32	32	32	32	18	18	164
11	32	10	19	14	10	32	117
12	10	19	42	32	10	32	145
13	32	19	19	42	18	18	148
14	32	32	42	42	42	10	200
15	10	32	14	18	18	32	124
16	32	32	10	18	42	14	148
17	19	19	14	19	19	32	122
18	32	19	42	18	32	14	157
19	10	19	19	32	32	32	144
20	32	42	10	32	42	14	172

Table A.42 Bootstrap Samples, Replication 2, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	14	18	10	42	42	42	168
2	18	14	42	18	19	18	129
3	18	18	19	18	19	14	106
4	10	14	14	14	42	14	108
5	14	19	42	32	42	19	168
6	42	18	32	32	42	14	180
7	19	32	14	32	18	19	134
8	18	14	14	10	14	42	112
9	18	19	18	42	18	19	134
10	32	32	32	32	18	18	164
11	32	10	19	14	10	32	117
12	10	19	42	32	10	32	145
13	32	19	19	42	18	18	148
14	32	32	42	42	42	10	200
15	10	32	14	18	18	32	124
16	32	32	10	18	42	14	148
17	19	19	14	19	19	32	122
18	32	19	42	18	32	14	157
19	10	19	19	32	32	32	144
20	32	42	10	32	42	14	172

Table A.43 Bootstrap Samples, Replication 3, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	42	10	42	14	19	42	169
2	19	32	32	14	19	18	134
3	42	19	10	14	42	42	169
4	32	42	10	32	14	18	148
5	42	14	18	32	32	10	148
6	10	19	32	32	42	18	153
7	32	42	18	10	42	42	186
8	10	10	42	14	19	32	127
9	14	42	32	42	14	19	163
10	18	18	10	10	14	10	80
11	10	42	10	18	18	10	108
12	32	18	19	32	14	14	129
13	14	19	19	32	10	42	136
14	10	10	10	10	14	18	72
15	10	10	32	19	32	10	113
16	10	32	42	18	18	32	152
17	42	19	18	19	10	10	118
18	18	19	32	10	19	10	108
19	19	19	42	42	19	14	155
20	10	42	32	42	10	42	178

Table A.44 Bootstrap Samples, Replication 4, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	19	19	42	32	18	42	172
2	14	18	18	19	32	32	133
3	14	14	10	14	32	19	103
4	42	32	32	18	32	42	198
5	19	14	18	42	42	10	145
6	18	32	32	42	19	42	185
7	19	19	18	42	19	42	159
8	18	14	18	42	10	10	112
9	42	42	14	18	19	32	167
10	32	32	18	18	14	19	133
11	19	32	19	14	14	42	140
12	18	19	14	18	18	18	105
13	32	14	10	18	42	18	134
14	19	18	14	10	42	19	122
15	19	19	32	42	18	42	172
16	42	19	10	19	32	10	132
17	10	18	14	32	14	42	130
18	18	32	42	18	10	18	138
19	10	42	42	10	19	32	155
20	32	32	19	32	42	10	167

Table A.45 Bootstrap Samples, Replication 5, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	14	10	42	18	10	14	108
2	42	42	42	18	32	10	186
3	32	32	14	42	32	14	166
4	14	14	18	32	10	32	120
5	14	32	42	42	18	32	180
6	14	19	32	42	42	10	159
7	19	14	18	10	14	32	107
8	18	32	42	19	10	32	153
9	42	14	10	14	32	42	154
10	14	18	19	32	42	19	144
11	19	42	10	32	42	19	164
12	19	42	14	18	42	18	153
13	14	18	18	32	10	18	110
14	42	10	42	14	14	10	132
15	18	19	42	10	19	19	127
16	18	42	42	32	14	19	167
17	14	14	14	32	14	10	98
18	32	10	18	10	32	14	116
19	18	18	19	10	19	42	126
20	14	32	42	19	18	19	144

Table A.46 Bootstrap Samples, Replication 6, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	10	10	32	18	19	10	99
2	10	14	14	42	32	18	130
3	19	14	19	14	10	19	95
4	32	14	32	19	10	18	125
5	42	32	18	42	14	18	166
6	19	10	19	32	14	32	126
7	18	18	42	10	32	18	138
8	18	14	14	42	42	14	144
9	32	32	19	19	14	32	148
10	18	42	42	10	18	10	140
11	42	10	42	42	10	19	165
12	19	42	32	32	19	18	162
13	14	42	32	19	19	10	136
14	18	32	18	42	42	19	171
15	42	32	32	18	42	19	185
16	42	18	19	42	14	14	149
17	32	14	18	10	32	32	138
18	14	32	42	32	10	19	149
19	10	14	42	42	18	10	136
20	42	32	10	42	42	14	182

Table A.47 Bootstrap Samples, Replication 7, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	19	18	18	18	18	32	123
2	32	42	19	18	18	19	148
3	14	42	19	42	10	32	159
4	42	32	10	14	10	19	127
5	32	14	19	19	42	10	136
6	10	42	19	10	14	32	127
7	42	14	42	32	42	42	214
8	32	42	14	10	42	19	159
9	10	18	14	42	10	10	104
10	19	18	19	18	42	14	130
11	42	42	32	32	32	10	190
12	14	14	19	14	42	14	117
13	42	19	42	18	10	32	163
14	18	42	32	18	18	18	146
15	19	18	18	18	19	14	106
16	42	32	14	14	19	42	163
17	42	19	42	19	32	42	196
18	10	19	14	32	18	19	112
19	10	14	18	10	32	18	102
20	19	19	10	14	19	10	91

Table A.48 Bootstrap Samples, Replication 8, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	19	19	14	42	18	14	126
2	10	10	42	18	10	18	108
3	10	32	19	14	19	18	112
4	10	10	14	32	10	42	118
5	19	10	18	14	42	19	122
6	14	19	14	32	14	42	135
7	19	18	32	14	14	18	115
8	19	19	19	18	14	42	131
9	19	14	14	32	14	42	135
10	18	10	19	14	42	10	113
11	14	18	14	19	42	19	126
12	14	10	19	42	10	14	109
13	14	18	10	14	18	19	93
14	32	42	32	32	18	42	198
15	18	32	18	42	42	10	162
16	32	10	18	18	32	42	152
17	10	19	19	19	18	19	104
18	10	19	42	32	14	14	131
19	19	32	42	18	18	19	148
20	14	42	14	10	19	19	118

Table A.49 Bootstrap Samples, Replication 9, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	18	42	10	18	42	18	148
2	42	18	18	19	19	42	158
3	14	42	19	19	10	18	122
4	32	32	42	42	14	10	172
5	18	14	18	18	14	19	101
6	19	18	14	32	18	32	133
7	14	19	14	18	42	18	125
8	32	32	32	32	42	19	189
9	42	32	19	19	10	42	164
10	42	32	42	18	32	19	185
11	32	32	32	10	18	14	138
12	19	18	18	19	19	32	125
13	19	14	42	19	10	32	136
14	32	14	19	14	18	19	116
15	42	10	19	32	32	14	149
16	19	19	42	18	10	42	150
17	19	10	42	10	42	19	142
18	14	42	42	10	19	32	159
19	42	19	32	19	14	18	144
20	18	18	19	42	19	10	126

Table A.50 Bootstrap Samples, Replication 10, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	19	42	10	32	18	10	131
2	19	42	18	10	18	14	121
3	18	18	42	19	32	32	161
4	42	10	10	14	18	42	136
5	10	10	19	14	14	18	85
6	19	18	10	18	14	10	89
7	14	32	32	10	19	32	139
8	32	18	42	14	32	10	148
9	19	19	14	10	14	32	108
10	14	19	42	18	10	18	121
11	10	10	10	19	14	10	73
12	32	19	14	42	10	32	149
13	18	10	18	32	14	14	106
14	18	19	42	14	18	42	153
15	10	42	18	42	19	42	173
16	18	19	19	18	10	32	116
17	32	14	10	18	19	42	135
18	18	14	18	14	18	19	101
19	14	19	42	18	32	19	144
20	14	14	32	10	18	19	107

Table A.51 Bootstrap Samples, Replication 11, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	10	32	42	18	42	18	162
2	10	14	42	14	32	32	144
3	10	10	32	32	18	10	112
4	32	42	19	18	14	42	167
5	10	14	42	18	14	18	116
6	32	14	32	42	18	14	152
7	32	32	18	10	19	42	153
8	19	19	18	18	10	10	94
9	18	32	18	19	10	19	116
10	10	10	18	18	42	14	112
11	42	18	42	18	32	14	166
12	32	18	10	32	19	18	129
13	32	18	18	42	10	10	130
14	14	14	32	18	10	19	107
15	14	10	42	32	32	18	148
16	18	19	14	18	19	18	106
17	32	18	18	18	10	14	110
18	42	32	14	10	14	32	144
19	32	19	14	10	14	32	121
20	19	14	32	32	14	42	153

Table A.52 Bootstrap Samples, Replication 12, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	32	18	18	18	10	10	106
2	14	10	14	18	14	10	80
3	32	19	19	10	32	42	154
4	19	32	42	32	19	18	162
5	19	19	19	32	18	10	117
6	14	42	19	19	19	19	132
7	19	18	18	14	10	10	89
8	42	10	18	14	42	18	144
9	18	42	32	42	32	32	198
10	14	10	19	19	42	14	118
11	18	19	42	10	42	14	145
12	14	19	10	18	19	32	112
13	19	18	32	19	18	42	148
14	19	42	10	14	18	18	121
15	42	42	10	10	18	14	136
16	14	42	42	18	32	32	180
17	14	19	19	14	42	42	150
18	10	18	18	18	19	18	101
19	18	19	19	32	14	32	134
20	19	42	10	19	42	10	142

Table A.53 Bootstrap Samples, Replication 13, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	14	10	18	10	32	19	103
2	42	14	42	42	14	32	186
3	14	19	42	14	19	14	122
4	32	19	14	10	18	10	103
5	18	42	10	42	18	18	148
6	10	19	10	18	14	10	81
7	18	14	42	32	32	18	156
8	14	14	10	14	19	18	89
9	19	32	32	18	42	18	161
10	32	19	14	32	14	32	143
11	18	10	42	14	10	10	104
12	32	19	42	19	42	10	164
13	32	14	14	19	18	42	139
14	10	32	42	14	14	10	122
15	10	32	10	10	14	14	90
16	32	42	14	18	10	32	148
17	14	18	14	14	19	32	111
18	32	19	10	42	42	14	159
19	19	18	10	10	18	14	89
20	19	42	14	19	10	18	122

Table A.54 Bootstrap Samples, Replication 14, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	14	32	19	32	10	19	126
2	14	42	10	10	32	42	150
3	10	10	10	19	32	10	91
4	32	10	18	10	10	19	99
5	14	19	32	10	19	18	112
6	14	14	18	32	42	19	139
7	42	42	18	14	10	19	145
8	14	18	10	19	18	14	93
9	10	42	32	19	10	14	127
10	19	10	42	18	19	32	140
11	42	42	32	32	19	14	181
12	14	42	18	32	14	32	152
13	10	14	32	19	19	42	136
14	32	14	42	32	18	14	152
15	14	14	32	18	42	18	138
16	14	18	14	42	19	42	149
17	18	42	32	14	19	18	143
18	19	18	32	14	19	18	120
19	32	18	18	32	42	18	160
20	19	19	18	42	19	14	131

Table A.55 Bootstrap Samples, Replication 15, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	14	10	32	42	18	42	158
2	19	32	18	10	32	19	130
3	32	18	19	42	32	18	161
4	32	14	32	14	10	18	120
5	42	19	10	14	10	42	137
6	32	42	32	42	42	18	208
7	32	18	42	32	19	42	185
8	10	32	19	18	14	42	135
9	19	19	10	10	18	10	86
10	18	10	14	14	32	18	106
11	19	19	14	19	32	42	145
12	14	18	19	18	42	10	121
13	18	42	18	18	32	10	138
14	10	32	14	32	10	42	140
15	10	32	18	10	10	14	94
16	32	10	10	18	14	18	102
17	14	19	32	42	18	18	143
18	14	10	14	42	19	32	131
19	19	14	42	42	10	32	159
20	18	32	18	32	10	19	129

Table A.56 Bootstrap Samples, Replication 16, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	19	14	42	42	14	10	141
2	32	18	42	42	42	42	218
3	14	42	42	10	14	18	140
4	18	19	18	19	18	14	106
5	14	10	18	19	32	18	111
6	32	32	19	10	18	14	125
7	42	14	18	18	32	32	156
8	14	14	19	10	19	19	95
9	19	10	18	14	14	18	93
10	19	18	32	14	18	10	111
11	42	18	42	18	18	32	170
12	19	32	14	32	19	19	135
13	42	10	42	19	32	32	177
14	19	19	32	42	32	14	158
15	42	19	19	10	19	19	128
16	18	32	32	14	14	42	152
17	18	18	42	42	10	42	172
18	42	19	10	19	19	10	119
19	18	42	18	14	32	18	142
20	32	10	19	42	14	19	136

Table A.57 Bootstrap Samples, Replication 17, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	32	14	10	10	14	10	90
2	14	14	19	42	42	10	141
3	14	18	42	42	42	32	190
4	19	19	42	18	18	18	134
5	32	19	32	19	10	10	122
6	14	18	10	18	14	18	92
7	18	10	14	18	14	10	84
8	10	14	18	18	18	42	120
9	32	14	32	32	14	42	166
10	42	32	32	10	10	19	145
11	18	32	42	42	10	19	163
12	32	32	32	14	10	10	130
13	19	10	18	19	18	14	98
14	10	18	32	14	10	18	102
15	14	42	18	32	18	14	138
16	32	42	42	10	10	10	146
17	19	18	18	19	32	14	120
18	32	14	19	42	32	19	158
19	14	42	42	14	10	10	132
20	32	32	32	42	14	19	171

Table A.58 Bootstrap Samples, Replication 18, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	19	32	19	19	19	32	140
2	18	32	32	42	14	19	157
3	10	42	32	18	14	14	130
4	32	18	42	42	19	42	195
5	10	14	42	19	18	32	135
6	10	19	42	32	42	32	177
7	18	19	18	42	32	14	143
8	42	19	19	19	10	32	141
9	42	18	14	32	42	18	166
10	14	10	10	42	18	42	136
11	14	32	14	19	10	32	121
12	19	14	19	10	19	42	123
13	32	18	19	14	42	19	144
14	14	42	10	42	19	10	137
15	19	19	32	32	14	42	158
16	14	18	18	42	19	10	121
17	14	14	10	32	19	42	131
18	10	42	10	42	14	18	136
19	19	32	42	10	32	32	167
20	10	42	32	10	42	19	155

Table A.59 Bootstrap Samples, Replication 19, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	32	32	42	18	32	32	188
2	10	10	19	14	14	14	81
3	19	18	10	14	32	32	125
4	32	14	32	10	10	42	140
5	32	32	32	14	32	32	174
6	19	10	18	14	14	32	107
7	19	42	10	42	32	10	155
8	32	42	10	10	14	14	122
9	32	32	42	10	10	18	144
10	19	10	32	42	18	19	140
11	14	32	32	14	10	42	144
12	18	14	42	42	10	18	144
13	14	42	32	10	42	19	159
14	10	14	19	19	14	19	95
15	42	19	18	42	32	32	185
16	19	18	42	19	10	10	118
17	18	14	19	32	42	32	157
18	18	32	32	32	19	32	165
19	18	14	14	42	14	42	144
20	18	10	10	32	19	19	108

Table A.60 Bootstrap Samples, Replication 20, Scenario 1 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	19	14	32	10	42	10	127
2	32	42	19	42	10	32	177
3	14	19	19	10	10	18	90
4	19	10	14	18	10	10	81
5	42	18	18	14	14	19	125
6	18	10	32	18	18	32	128
7	42	18	10	18	42	19	149
8	32	18	18	42	18	10	138
9	32	32	42	42	42	10	200
10	18	32	42	10	32	19	153
11	14	19	14	14	10	18	89
12	32	32	10	42	14	10	140
13	18	10	10	10	19	42	109
14	14	14	18	18	19	19	102
15	18	14	32	14	32	19	129
16	18	32	19	42	18	19	148
17	10	32	14	19	32	32	139
18	14	18	42	19	10	14	117
19	10	32	14	42	32	19	149
20	14	18	19	19	18	18	106

A.4 Scenario 1 U-Boat Kills

Table A.61 Bootstrap Samples, Replication 1, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	0	1	0	1	1	3
2	1	1	1	1	0	1	5
3	1	0	0	1	0	0	2
4	1	0	1	1	0	1	4
5	0	1	1	1	0	0	3
6	0	0	1	1	0	1	3
7	0	1	1	1	0	1	4
8	0	1	1	0	1	1	4
9	1	1	1	1	1	1	6
10	0	1	0	1	0	0	2
11	0	0	1	1	1	1	4
12	1	0	1	1	1	1	5
13	0	0	0	1	1	1	3
14	0	1	0	1	1	1	4
15	1	0	1	1	0	0	3
16	0	0	1	0	0	1	2
17	1	1	0	1	1	1	5
18	0	1	1	1	0	0	3
19	1	0	1	0	0	1	3
20	0	0	1	1	0	1	3

Table A.62 Bootstrap Samples, Replication 2, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	1	1	1	1	0	5
2	0	1	0	0	1	0	2
3	0	1	0	1	1	0	3
4	0	1	1	1	0	1	4
5	1	0	1	0	1	1	4
6	1	0	0	1	0	0	2
7	0	1	0	1	0	0	2
8	1	1	0	1	1	0	4
9	0	0	0	1	1	0	2
10	0	1	0	0	1	0	2
11	1	0	1	0	1	0	3
12	1	1	0	1	1	0	4
13	1	0	1	1	0	1	4
14	1	1	0	0	1	1	4
15	0	1	1	0	0	0	2
16	1	0	1	0	1	1	4
17	0	1	1	0	0	0	2
18	0	1	1	0	0	0	2
19	0	1	1	0	0	0	2
20	1	0	1	1	1	0	4

Table A.63 Bootstrap Samples, Replication 3, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	1	1	0	0	0	2
2	0	0	0	1	0	0	1
3	1	1	1	0	1	0	4
4	1	1	1	0	0	1	4
5	0	0	0	1	0	1	2
6	1	1	0	1	0	1	4
7	1	0	0	1	0	1	3
8	1	1	0	0	1	0	3
9	0	0	1	1	1	0	3
10	0	1	1	1	1	1	5
11	1	1	0	1	1	0	4
12	0	0	0	1	0	1	2
13	0	1	0	1	0	1	3
14	0	0	1	0	1	0	2
15	0	1	0	0	1	1	3
16	1	0	0	1	0	0	2
17	1	0	1	1	1	0	4
18	1	1	1	1	0	1	5
19	1	1	1	0	0	0	3
20	0	0	0	1	0	1	2

Table A.64 Bootstrap Samples, Replication 4, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	1	0	0	0	1	3
2	1	0	0	0	0	1	2
3	0	1	0	0	1	1	3
4	1	0	0	1	1	1	4
5	0	0	1	1	0	0	2
6	0	0	0	1	1	1	3
7	1	1	0	1	0	1	4
8	0	1	1	1	1	1	5
9	1	1	0	0	0	0	2
10	1	0	1	1	0	1	4
11	1	0	1	1	0	1	4
12	1	1	0	1	0	1	4
13	0	1	1	1	1	0	4
14	0	0	0	0	1	1	2
15	1	1	0	1	0	1	4
16	0	1	1	1	0	0	3
17	0	0	0	0	1	0	1
18	0	1	0	0	1	0	2
19	1	1	1	1	0	1	5
20	0	0	0	0	0	0	0

Table A.65 Bootstrap Samples, Replication 5, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	0	1	1	1	1	4
2	1	0	1	1	1	0	4
3	1	0	1	0	1	1	4
4	1	0	1	1	0	1	4
5	1	0	1	1	0	0	3
6	1	1	0	1	1	0	4
7	0	1	1	1	0	1	4
8	0	1	1	1	1	1	5
9	1	0	1	1	1	0	4
10	1	1	0	1	0	1	4
11	0	0	1	0	1	1	3
12	1	0	0	0	1	0	2
13	1	0	0	1	0	1	3
14	1	1	0	1	1	0	4
15	0	1	0	1	0	0	2
16	1	1	1	1	1	1	6
17	0	1	0	0	0	1	2
18	1	1	1	1	0	0	4
19	0	1	0	0	1	0	2
20	1	0	0	1	0	0	2

Table A.66 Bootstrap Samples, Replication 6, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	1	0	1	1	1	5
2	0	1	1	0	0	1	3
3	0	0	1	1	0	0	2
4	0	1	1	1	1	0	4
5	1	0	0	1	1	0	3
6	1	1	1	0	1	0	4
7	0	1	1	0	0	1	3
8	1	0	1	0	1	1	4
9	1	0	0	1	1	0	3
10	1	0	0	1	1	0	3
11	1	1	0	0	1	1	4
12	0	0	0	1	1	0	2
13	0	1	0	1	1	1	4
14	1	1	1	0	1	0	4
15	0	0	0	0	0	0	0
16	0	1	0	0	1	0	2
17	0	0	1	1	0	1	3
18	1	1	0	1	1	0	4
19	1	0	1	1	1	0	4
20	0	1	0	0	1	1	3

Table A.67 Bootstrap Samples, Replication 7, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	0	1	1	0	0	2
2	0	1	0	1	1	0	3
3	1	1	0	0	1	0	3
4	1	0	1	1	1	1	5
5	1	1	1	1	0	1	5
6	1	1	1	1	0	0	4
7	0	0	0	1	0	0	1
8	1	1	1	0	0	0	3
9	1	0	0	0	1	1	3
10	0	0	0	0	1	0	1
11	1	0	1	1	1	0	4
12	0	0	0	0	0	0	0
13	1	1	1	0	0	0	3
14	1	1	1	1	1	1	6
15	1	0	0	0	0	0	1
16	0	0	0	0	0	1	1
17	0	1	0	0	0	0	1
18	1	1	1	1	1	0	5
19	0	1	1	0	1	1	4
20	0	1	1	1	1	0	4

Table A.68 Bootstrap Samples, Replication 8, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	1	0	1	1	1	4
2	1	0	1	1	1	1	5
3	1	1	1	1	1	0	5
4	0	0	0	0	1	0	1
5	0	1	1	0	1	1	4
6	1	1	0	0	1	0	3
7	1	0	0	0	1	1	3
8	1	0	0	0	0	1	2
9	1	1	1	1	1	1	6
10	0	0	0	0	1	1	2
11	0	0	1	0	0	0	1
12	0	1	1	0	0	0	2
13	0	1	1	1	1	1	5
14	0	0	1	1	1	0	3
15	0	0	0	1	1	0	2
16	0	0	0	1	1	0	2
17	1	1	0	1	1	0	4
18	0	1	1	1	0	0	3
19	0	1	1	0	0	0	2
20	0	1	0	0	1	0	2

Table A.69 Bootstrap Samples, Replication 9, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	1	1	0	1	0	3
2	0	1	0	1	1	1	4
3	1	1	1	1	0	0	4
4	1	1	1	0	0	1	4
5	0	1	1	1	0	1	4
6	0	1	1	1	0	0	3
7	0	0	0	1	0	1	2
8	1	1	1	0	1	1	5
9	1	1	1	1	0	0	4
10	0	1	1	1	0	0	3
11	0	0	1	0	0	1	2
12	1	0	0	1	1	0	3
13	0	1	1	0	1	0	3
14	0	1	1	1	0	0	3
15	0	0	0	0	1	0	1
16	0	0	1	0	0	0	1
17	0	1	1	0	1	0	3
18	1	0	0	0	1	1	3
19	1	0	1	1	1	1	5
20	1	1	0	0	0	1	3

Table A.70 Bootstrap Samples, Replication 10, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	1	1	0	1	1	5
2	0	1	1	1	0	0	3
3	0	0	0	0	0	1	1
4	1	0	0	1	0	1	3
5	1	0	1	1	1	0	4
6	1	0	0	1	1	1	4
7	1	1	1	1	1	1	6
8	1	0	1	1	0	0	3
9	1	1	0	1	1	0	4
10	0	0	1	1	0	1	3
11	1	0	1	0	0	1	3
12	1	1	0	1	1	1	5
13	0	0	0	1	1	1	3
14	1	1	0	0	1	1	4
15	1	1	0	1	1	1	5
16	1	1	0	1	0	0	3
17	0	1	0	0	0	1	2
18	1	0	1	1	0	1	4
19	0	1	1	0	1	0	3
20	0	0	1	0	1	0	2

Table A.71 Bootstrap Samples, Replication 11, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	0	1	1	1	0	3
2	0	1	0	0	0	0	1
3	0	1	1	0	1	0	3
4	0	0	1	1	1	0	3
5	0	1	1	1	0	0	3
6	1	0	0	0	1	1	3
7	1	1	1	0	1	0	4
8	0	1	1	1	1	1	5
9	0	0	1	1	1	1	4
10	1	1	1	0	1	1	5
11	0	1	0	0	0	1	2
12	1	1	0	1	1	1	5
13	1	0	0	1	1	0	3
14	1	1	1	1	1	0	5
15	1	0	1	0	1	0	3
16	1	1	0	0	0	0	2
17	0	0	0	0	1	1	2
18	0	1	1	1	1	1	5
19	0	0	0	1	1	1	3
20	1	0	1	1	1	1	5

Table A.72 Bootstrap Samples, Replication 12, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	1	0	1	0	0	3
2	1	1	1	1	0	0	4
3	1	0	1	0	1	0	3
4	1	0	1	0	0	0	2
5	0	1	1	1	0	1	4
6	1	1	1	1	1	1	6
7	1	0	1	0	1	1	4
8	0	0	0	0	1	0	1
9	1	0	1	1	1	1	5
10	0	0	1	0	1	0	2
11	0	0	1	1	1	0	3
12	0	0	1	1	0	0	2
13	0	0	1	0	1	0	2
14	0	0	1	1	0	1	3
15	1	0	1	0	1	0	3
16	1	1	0	0	1	0	3
17	0	1	1	1	0	1	4
18	1	1	0	0	1	1	4
19	1	0	0	1	1	1	4
20	0	1	1	0	0	1	3

Table A.73 Bootstrap Samples, Replication 13, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	0	1	0	1	1	4
2	1	1	1	1	0	1	5
3	0	1	1	1	0	0	3
4	0	1	1	0	0	1	3
5	0	1	1	0	1	1	4
6	1	0	0	1	0	1	3
7	1	1	1	0	0	0	3
8	0	1	0	1	1	1	4
9	0	1	0	0	1	0	2
10	1	0	1	1	0	0	3
11	0	1	1	1	0	1	4
12	1	1	1	0	1	0	4
13	0	1	1	0	0	0	2
14	0	1	0	1	1	0	3
15	0	1	0	1	0	0	2
16	0	1	1	0	0	1	3
17	0	0	0	1	0	1	2
18	1	1	0	1	1	0	4
19	1	1	1	1	0	1	5
20	0	1	1	0	1	1	4

Table A.74 Bootstrap Samples, Replication 14, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	1	0	1	0	0	2
2	0	1	0	0	1	1	3
3	1	0	0	0	0	1	2
4	0	0	1	1	0	0	2
5	1	0	1	0	1	0	3
6	0	1	1	0	1	0	3
7	0	1	1	0	1	1	4
8	1	1	1	0	1	0	4
9	0	1	0	1	1	1	4
10	1	0	0	1	0	1	3
11	1	0	0	1	0	0	2
12	0	1	1	1	0	1	4
13	0	1	0	1	0	0	2
14	0	1	1	1	1	1	5
15	1	1	0	1	1	1	5
16	0	0	0	0	1	1	2
17	1	1	0	1	0	0	3
18	0	0	1	1	0	0	2
19	0	0	1	1	1	0	3
20	1	1	1	0	0	0	3

Table A.75 Bootstrap Samples, Replication 15, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	1	0	1	1	1	5
2	1	1	1	1	1	1	6
3	0	1	1	1	0	1	4
4	0	0	1	1	0	0	2
5	1	0	1	1	0	1	4
6	1	1	0	0	1	0	3
7	0	1	0	1	1	1	4
8	0	0	0	0	1	1	2
9	1	0	1	1	0	1	4
10	1	0	1	1	0	1	4
11	0	0	0	0	1	0	1
12	1	1	1	1	1	0	5
13	1	0	1	1	0	0	3
14	0	0	1	0	1	1	3
15	1	0	1	0	0	0	2
16	1	1	1	1	0	1	5
17	1	0	1	0	1	1	4
18	0	1	1	0	1	0	3
19	1	1	0	1	0	1	4
20	1	1	1	1	1	0	5

Table A.76 Bootstrap Samples, Replication 16, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	0	1	1	0	0	3
2	0	0	0	0	0	0	0
3	1	0	0	1	1	0	3
4	0	0	1	1	1	1	4
5	0	1	0	1	1	1	4
6	1	0	1	0	0	1	3
7	0	1	0	0	1	1	3
8	0	0	0	1	0	1	2
9	0	1	1	1	1	0	4
10	1	1	1	0	0	1	4
11	1	1	0	1	1	1	5
12	1	0	1	0	1	0	3
13	1	1	1	1	0	1	5
14	0	0	0	0	0	1	1
15	0	0	0	0	0	1	1
16	1	0	0	1	1	1	4
17	0	0	0	0	1	0	1
18	1	1	1	1	1	1	6
19	1	0	0	0	0	0	1
20	1	0	1	0	1	1	4

Table A.77 Bootstrap Samples, Replication 17, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	1	1	0	0	0	3
2	1	0	0	1	1	0	3
3	1	1	1	1	0	0	4
4	0	1	0	1	0	1	3
5	0	0	0	1	0	1	2
6	0	1	0	1	0	0	2
7	1	0	1	1	1	0	4
8	0	1	1	0	1	0	3
9	1	1	1	0	1	1	5
10	1	1	0	1	1	1	5
11	0	1	1	1	1	1	5
12	0	1	0	0	0	0	1
13	0	1	0	1	1	0	3
14	1	0	1	0	0	0	2
15	0	1	1	1	1	1	5
16	0	0	1	1	1	0	3
17	1	0	1	1	0	1	4
18	1	1	1	1	0	1	5
19	0	1	1	1	0	1	4
20	0	0	0	0	1	1	2

Table A.78 Bootstrap Samples, Replication 18, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	1	1	1	0	0	3
2	0	0	0	1	0	0	1
3	0	1	1	0	0	1	3
4	1	0	1	0	1	1	4
5	1	1	1	0	1	0	4
6	0	1	1	1	1	1	5
7	1	1	1	0	0	0	3
8	1	0	1	0	1	1	4
9	1	1	0	1	0	0	3
10	0	0	1	1	1	0	3
11	1	1	1	0	1	0	4
12	0	1	0	1	1	0	3
13	1	1	0	1	1	1	5
14	1	1	0	0	1	1	4
15	1	1	0	0	1	0	3
16	0	1	1	1	0	0	3
17	0	1	1	0	1	1	4
18	0	1	1	1	1	1	5
19	1	1	1	0	0	0	3
20	1	1	0	1	1	0	4

Table A.79 Bootstrap Samples, Replication 19, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	0	1	1	1	0	3
2	1	0	1	0	0	0	2
3	0	0	0	1	0	1	2
4	1	1	0	1	0	0	3
5	0	1	1	0	1	0	3
6	1	1	1	0	1	1	5
7	1	1	0	1	1	0	4
8	0	0	1	1	0	1	3
9	0	1	1	1	1	0	4
10	0	0	1	0	1	1	3
11	0	1	0	0	0	1	2
12	1	0	0	0	0	0	1
13	0	1	0	0	1	1	3
14	1	1	0	1	1	1	5
15	1	0	0	0	0	0	1
16	0	0	1	1	0	0	2
17	1	0	1	1	0	1	4
18	0	0	1	0	0	0	1
19	0	1	1	0	1	0	3
20	0	1	0	1	1	1	4

Table A.80 Bootstrap Samples, Replication 20, Scenario 1 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	0	0	0	1	1	1	3
2	1	1	0	1	1	1	5
3	1	1	1	0	0	1	4
4	1	1	0	1	1	0	4
5	0	1	1	1	0	1	4
6	1	0	0	0	0	0	1
7	0	1	1	0	1	0	3
8	1	1	0	0	0	1	3
9	1	1	0	1	1	0	4
10	1	1	1	0	1	0	4
11	0	1	0	0	0	0	1
12	1	1	1	1	0	1	5
13	0	1	0	0	0	0	1
14	0	0	1	0	1	1	3
15	0	0	1	0	1	0	2
16	0	0	0	0	0	1	1
17	1	1	1	0	1	0	4
18	1	1	0	0	0	1	3
19	0	1	1	0	0	0	2
20	0	0	1	1	1	1	4

A.5 Scenario 2 U-Boat Sightings

Table A.81 Bootstrap Samples, Replication 1, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	81	7	52	60	98	52	350
2	98	98	21	98	81	98	494
3	98	81	81	21	60	7	348
4	98	7	52	52	60	52	321
5	81	52	52	52	60	60	357
6	81	81	98	52	7	52	371
7	60	98	98	21	7	21	305
8	7	52	98	81	21	98	357
9	52	52	52	52	21	98	327
10	60	98	60	52	81	60	411
11	81	81	21	21	52	98	354
12	98	60	21	52	52	21	304
13	60	7	81	52	21	52	273
14	7	52	60	52	21	52	244
15	52	81	98	21	81	81	414
16	7	81	21	60	81	52	302
17	98	52	7	21	21	21	220
18	60	98	98	21	7	60	344
19	52	60	21	81	81	98	393
20	7	81	98	21	81	21	309

Table A.82 Bootstrap Samples, Replication 2, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	81	52	7	7	52	81	280
2	52	98	52	98	60	98	458
3	81	21	98	81	7	98	386
4	52	60	60	60	98	98	428
5	52	7	52	52	60	52	275
6	98	98	81	81	52	52	462
7	52	21	60	60	81	52	326
8	52	60	81	98	21	7	319
9	52	98	21	60	81	81	393
10	98	81	81	60	98	60	478
11	7	7	7	60	98	52	231
12	98	98	60	52	7	98	413
13	81	7	52	81	7	7	235
14	60	21	21	52	81	7	242
15	60	81	60	81	52	21	355
16	60	52	21	60	98	98	389
17	7	52	52	52	7	21	191
18	81	81	60	21	81	81	405
19	21	7	98	21	60	52	259
20	52	81	98	60	98	52	441

Table A.83 Bootstrap Samples, Replication 3, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	98	7	7	81	52	60	305
2	52	60	52	98	81	98	441
3	98	52	21	52	52	98	373
4	81	52	7	7	81	7	235
5	81	7	21	21	60	21	211
6	81	60	52	52	21	21	287
7	21	60	52	7	7	98	245
8	60	7	60	81	98	60	366
9	21	98	52	52	81	98	402
10	7	81	98	98	81	81	446
11	81	81	7	21	98	52	340
12	21	60	81	7	21	52	242
13	7	21	98	98	7	52	283
14	52	52	7	60	60	81	312
15	52	21	81	60	98	60	372
16	60	52	81	52	7	52	304
17	81	81	7	7	60	52	288
18	52	98	21	7	98	98	374
19	98	7	21	7	60	21	214
20	98	7	52	21	52	98	328

Table A.84 Bootstrap Samples, Replication 4, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	81	81	21	21	81	7	292
2	52	21	52	60	52	81	318
3	98	98	81	60	98	7	442
4	98	60	7	98	21	7	291
5	81	52	52	21	60	60	326
6	7	52	7	7	60	7	140
7	7	98	7	81	81	52	326
8	21	60	52	98	21	60	312
9	98	98	52	7	21	7	283
10	7	7	52	81	60	60	267
11	81	98	52	60	52	7	350
12	21	7	81	81	21	52	263
13	21	52	21	7	52	98	251
14	21	21	81	60	21	98	302
15	98	60	81	81	98	60	478
16	60	60	7	98	60	7	292
17	60	60	98	81	60	81	440
18	7	7	98	60	60	98	330
19	98	81	7	81	81	52	400
20	21	52	98	7	81	81	340

Table A.84 Bootstrap Samples, Replication 4, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	98	60	81	81	81	60	461
2	7	60	60	7	98	21	253
3	21	60	7	21	7	52	168
4	21	60	98	21	7	98	305
5	60	81	21	81	98	7	348
6	52	60	98	21	81	98	410
7	21	21	7	81	81	60	271
8	21	7	81	60	7	21	197
9	52	52	60	21	21	52	258
10	52	7	60	52	21	52	244
11	21	52	81	98	52	52	356
12	21	81	98	52	98	98	448
13	81	60	21	98	21	98	379
14	98	60	98	81	7	60	404
15	7	60	21	7	98	81	274
16	60	52	81	52	81	98	424
17	52	52	21	81	81	21	308
18	98	21	60	98	60	7	344
19	7	52	60	21	81	81	302
20	98	60	98	81	7	21	365

Table A.85 Bootstrap Samples, Replication 5, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	81	81	21	60	21	7	271
2	21	98	98	98	98	52	465
3	81	81	81	81	98	60	482
4	60	7	7	98	52	60	284
5	60	98	81	98	21	7	365
6	98	7	7	52	81	98	343
7	60	21	7	98	21	60	267
8	98	60	21	81	98	60	418
9	60	7	7	7	81	7	169
10	7	52	60	81	81	52	333
11	98	7	52	98	98	52	405
12	60	81	21	21	21	60	264
13	21	98	81	21	98	7	326
14	52	60	52	98	52	60	374
15	98	81	81	52	81	60	453
16	60	21	60	81	52	52	326
17	52	52	81	60	60	21	326
18	60	52	7	52	81	60	312
19	52	81	7	81	52	7	280
20	98	60	60	81	7	21	327

Table A.86 Bootstrap Samples, Replication 6, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	81	81	21	7	21	98	309
2	98	81	7	21	60	7	274
3	81	81	7	21	21	60	271
4	52	98	81	60	7	7	305
5	21	81	52	52	98	7	311
6	52	21	52	81	52	60	318
7	98	60	98	21	7	60	344
8	81	7	52	98	98	52	388
9	60	81	7	7	98	52	305
10	98	98	21	81	52	7	357
11	7	81	81	60	21	60	310
12	52	81	60	52	81	60	386
13	52	98	52	52	7	52	313
14	52	81	7	98	52	52	342
15	81	81	52	52	52	52	370
16	21	21	21	98	7	81	249
17	52	81	60	81	60	81	415
18	52	98	81	7	21	52	311
19	7	60	52	98	98	52	367
20	52	52	52	60	60	81	357

Table A.87 Bootstrap Samples, Replication 7, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	21	7	98	52	81	21	280
2	52	60	21	98	7	60	298
3	21	81	98	98	81	60	439
4	98	98	81	7	81	98	463
5	52	52	98	7	60	21	290
6	60	52	81	60	7	52	312
7	60	21	60	52	7	52	252
8	7	81	52	98	52	21	311
9	81	21	81	7	21	98	309
10	52	52	7	81	21	81	294
11	21	21	98	7	52	98	297
12	98	60	81	60	81	52	432
13	81	60	52	21	52	98	364
14	7	98	52	81	98	60	396
15	52	98	81	98	21	52	402
16	60	7	98	81	81	52	379
17	81	81	98	60	21	98	439
18	98	98	52	98	7	98	451
19	21	98	81	21	81	7	309
20	98	81	7	21	52	7	266

Table A.88 Bootstrap Samples, Replication 8, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	21	21	60	81	52	60	295
2	98	98	52	98	60	52	458
3	52	21	21	98	52	52	296
4	52	21	52	21	52	52	250
5	21	60	21	7	7	7	123
6	21	60	52	52	52	60	297
7	60	7	60	21	81	21	250
8	7	81	7	52	98	52	297
9	21	21	81	81	98	81	383
10	81	98	98	7	52	21	357
11	98	21	52	7	7	98	283
12	60	60	98	52	52	60	382
13	81	81	52	52	21	60	347
14	21	7	81	21	98	21	249
15	60	7	7	60	81	52	267
16	7	7	7	21	81	98	221
17	98	98	52	98	98	98	542
18	52	21	81	98	21	81	354
19	21	81	81	98	7	98	386
20	81	21	7	7	52	60	228

Table A.89 Bootstrap Samples, Replication 9, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	98	52	52	98	60	98	458
2	60	21	81	52	7	52	273
3	81	81	52	52	21	81	368
4	52	52	98	7	81	21	311
5	21	81	98	60	81	60	401
6	60	52	7	52	7	21	199
7	21	7	52	81	21	52	234
8	7	21	21	81	81	7	218
9	21	21	60	7	81	98	288
10	81	7	21	21	7	21	158
11	21	21	81	7	7	52	189
12	60	98	81	21	52	21	333
13	21	81	52	60	60	98	372
14	7	60	81	98	98	81	425
15	60	52	52	60	52	7	283
16	81	60	60	81	98	21	401
17	98	98	98	21	52	60	427
18	81	21	81	81	60	60	384
19	21	7	60	98	21	98	305
20	21	21	7	81	21	7	158

Table A.90 Bootstrap Samples, Replication 10, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	60	98	98	7	7	7	277
2	60	81	60	21	52	52	326
3	21	52	81	7	7	21	189
4	81	52	60	52	81	7	333
5	7	81	52	7	52	21	220
6	52	21	60	7	60	81	281
7	98	60	60	81	7	60	366
8	7	81	52	21	21	21	203
9	52	60	21	98	21	98	350
10	81	52	52	60	81	7	333
11	52	7	21	81	21	52	234
12	21	52	7	52	81	98	311
13	60	81	52	52	60	81	386
14	81	52	81	81	52	81	428
15	21	81	81	60	52	81	376
16	52	98	81	60	52	7	350
17	21	52	7	60	52	98	290
18	7	81	60	52	98	81	379
19	7	98	98	21	98	81	403
20	7	98	52	52	60	81	350

Table A.91 Bootstrap Samples, Replication 11, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	21	60	7	52	7	98	245
2	7	60	7	81	7	7	169
3	21	21	52	7	60	98	259
4	7	81	52	52	21	98	311
5	60	81	7	98	21	7	274
6	52	21	98	98	81	98	448
7	7	98	60	81	52	21	319
8	81	60	21	21	98	60	341
9	81	98	7	7	7	21	221
10	21	60	98	52	98	52	381
11	52	52	21	60	81	60	326
12	98	7	7	21	7	21	161
13	52	7	98	98	81	21	357
14	60	60	81	60	98	81	440
15	98	52	81	52	52	7	342
16	21	60	81	98	98	7	365
17	7	7	52	21	60	7	154
18	7	21	52	60	60	21	221
19	7	98	98	52	60	98	413
20	21	21	60	98	52	7	259

Table A.93 Bootstrap Samples, Replication 13, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	60	81	60	52	7	81	341
2	60	81	52	98	21	52	364
3	7	98	21	60	21	81	288
4	21	52	98	81	7	81	340
5	81	81	7	21	98	98	386
6	7	60	60	52	98	81	358
7	60	21	60	98	81	52	372
8	21	98	98	98	7	98	420
9	98	81	21	7	98	81	386
10	81	52	21	7	60	21	242
11	98	81	21	81	52	7	340
12	52	7	98	52	60	52	321
13	21	60	52	21	60	52	266
14	98	98	7	7	81	21	312
15	60	52	21	7	7	7	154
16	81	60	81	7	52	7	288
17	7	7	98	81	7	21	221
18	52	7	21	52	60	7	199
19	7	52	21	81	21	21	203
20	21	52	7	98	52	52	282

Table A.94 Bootstrap Samples, Replication 14, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	98	21	21	98	60	52	350
2	98	98	21	7	52	98	374
3	81	7	81	60	60	81	370
4	52	52	21	7	7	52	191
5	98	52	60	60	21	60	351
6	52	52	52	52	21	7	236
7	7	21	21	98	7	7	161
8	98	98	81	60	21	21	379
9	21	21	60	98	52	98	350
10	81	52	52	81	21	81	368
11	21	60	98	81	52	98	410
12	52	98	7	81	7	81	326
13	81	21	7	60	7	60	236
14	98	21	60	21	21	60	281
15	81	98	21	52	60	21	333
16	60	21	21	7	21	52	182
17	21	21	98	21	60	98	319
18	98	52	98	52	81	7	388
19	21	60	21	21	21	7	151
20	60	52	60	81	60	98	411

Table A.95 Bootstrap Samples, Replication 15, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	98	81	81	81	60	81	482
2	98	81	60	52	81	21	393
3	21	21	7	7	7	81	144
4	52	98	7	7	7	81	252
5	60	98	98	60	7	52	375
6	60	98	81	21	98	98	456
7	52	21	81	60	98	60	372
8	21	81	7	52	21	60	242
9	81	21	81	21	60	52	316
10	52	52	7	98	7	52	268
11	21	60	98	7	60	98	344
12	98	52	81	98	52	98	479
13	7	7	60	60	21	7	162
14	98	60	60	81	52	52	403
15	81	98	21	7	81	98	386
16	52	60	60	98	60	21	351
17	98	52	98	60	98	81	487
18	98	60	7	98	52	52	367
19	52	60	52	7	81	81	333
20	60	21	7	21	7	21	137

Table A.96 Bootstrap Samples, Replication 16, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	98	21	7	81	98	7	312
2	52	60	60	52	52	98	374
3	98	7	98	81	7	52	343
4	52	52	60	81	81	52	378
5	52	52	7	81	7	21	220
6	60	81	60	7	60	7	275
7	81	21	98	60	7	60	327
8	7	52	98	52	98	98	405
9	60	98	60	21	98	81	418
10	81	60	60	81	60	21	363
11	81	52	60	7	60	81	341
12	98	21	81	52	7	7	266
13	21	52	60	52	98	81	364
14	52	98	60	81	60	21	372
15	98	21	81	98	21	60	379
16	21	52	81	52	60	7	273
17	81	81	60	60	98	81	461
18	81	98	21	7	7	52	266
19	52	98	81	98	81	21	431
20	52	81	60	7	81	81	362

Table A.97 Bootstrap Samples, Replication 17, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7	52	52	52	21	52	236
2	21	21	21	21	81	7	172
3	98	7	60	98	7	98	368
4	98	52	21	52	21	21	265
5	81	60	7	52	98	52	350
6	98	52	52	21	81	98	402
7	81	60	81	98	98	7	425
8	98	98	81	7	52	21	357
9	98	21	21	21	98	98	357
10	60	52	21	60	52	52	297
11	52	7	52	98	7	7	223
12	7	52	21	60	21	60	221
13	7	7	52	52	21	81	220
14	21	21	98	52	81	21	294
15	52	98	7	81	52	81	371
16	98	98	7	21	98	7	329
17	52	21	52	81	21	81	308
18	21	21	81	60	52	21	256
19	21	81	52	81	98	52	385
20	98	98	60	21	7	60	344

Table A.98 Bootstrap Samples, Replication 18, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	52	98	81	52	81	60	424
2	7	7	52	81	21	52	220
3	60	52	7	81	98	60	358
4	60	7	52	52	7	81	259
5	81	81	60	98	21	98	439
6	52	52	21	81	98	21	325
7	21	21	21	52	81	81	277
8	60	52	21	52	60	98	343
9	52	81	81	7	81	60	362
10	7	7	52	52	21	7	146
11	98	21	60	60	98	52	389
12	98	81	81	21	21	60	362
13	98	21	21	21	60	21	242
14	21	21	81	98	52	60	333
15	98	7	21	7	81	81	295
16	21	21	52	98	60	52	304
17	7	98	60	60	81	60	366
18	52	52	52	21	21	60	258
19	81	21	7	52	81	81	323
20	60	81	98	52	98	60	449

Table A.99 Bootstrap Samples, Replication 19, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	60	21	52	52	81	60	326
2	81	98	21	52	60	21	333
3	7	7	52	98	7	21	192
4	81	98	60	52	52	21	364
5	81	7	60	81	52	21	302
6	81	81	60	81	7	52	362
7	98	52	7	81	98	60	396
8	60	7	81	98	21	60	327
9	21	60	21	7	7	81	197
10	52	60	52	81	21	81	347
11	7	98	7	7	52	81	252
12	98	81	52	60	81	21	393
13	21	21	7	60	7	60	176
14	60	52	81	60	81	81	415
15	7	52	81	52	7	52	251
16	52	52	7	21	21	98	251
17	60	98	7	60	52	81	358
18	7	52	81	21	21	21	203
19	81	52	98	60	98	98	487
20	98	60	81	60	7	98	404

Table A.100 Bootstrap Samples, Replication 20, Scenario 2 U-Boat Sightings

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	60	60	60	98	7	81	366
2	7	60	98	21	21	60	267
3	21	21	60	52	60	7	221
4	60	52	60	98	60	21	351
5	81	98	21	7	21	7	235
6	52	21	98	81	52	98	402
7	60	52	60	60	21	7	260
8	98	60	81	7	52	7	305
9	60	60	98	81	7	81	387
10	52	52	98	60	98	98	458
11	98	7	21	60	81	21	288
12	60	21	52	81	81	98	393
13	7	7	60	81	7	81	243
14	81	81	98	21	60	7	348
15	21	21	7	21	21	52	143
16	52	52	60	60	98	21	343
17	98	21	7	21	7	81	235
18	21	52	60	21	98	21	273
19	98	98	98	7	81	81	463
20	60	60	81	52	7	7	267

A.6 Scenario 2 U-Boat Kills

Table A.101 Bootstrap Samples, Replication 1, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4	4	1	2	1	13	25
2	4	13	1	13	5	2	38
3	4	4	1	5	7	2	23
4	1	2	7	5	2	13	30
5	2	7	1	1	4	1	16
6	7	1	5	1	2	5	21
7	2	4	1	5	1	13	26
8	1	5	1	5	7	4	23
9	13	5	5	7	5	7	42
10	13	13	5	1	5	5	42
11	4	1	1	2	1	2	11
12	1	7	1	1	1	2	13
13	13	5	13	1	2	1	35
14	13	4	2	5	2	1	27
15	2	7	13	4	13	13	52
16	4	1	5	13	13	1	37
17	13	2	13	13	1	1	43
18	4	7	13	5	1	7	37
19	4	4	5	7	2	7	29
20	5	7	7	7	7	13	46

Table A.102 Bootstrap Samples, Replication 2, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	13	7	1	13	2	5	41
2	4	2	1	1	2	2	12
3	4	7	13	2	1	5	32
4	1	2	5	4	1	4	17
5	4	1	2	5	13	5	30
6	2	13	5	13	2	7	42
7	4	7	4	2	2	4	23
8	2	13	5	5	1	4	30
9	13	1	1	2	5	13	35
10	1	5	7	7	2	4	26
11	13	13	2	5	5	2	40
12	5	1	7	2	7	5	27
13	5	5	7	5	4	5	31
14	1	1	2	5	5	7	21
15	13	13	7	4	2	7	46
16	13	5	7	5	1	7	38
17	2	1	2	4	13	13	35
18	4	7	4	2	4	5	26
19	2	5	13	5	4	5	34
20	13	1	7	5	5	1	32

Table A.103 Bootstrap Samples, Replication 3, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	2	2	5	5	13	4	31
2	7	1	7	1	13	1	30
3	7	7	1	4	13	13	45
4	7	5	1	13	4	2	32
5	7	4	5	7	4	2	29
6	7	13	4	13	13	13	63
7	2	13	13	13	7	4	52
8	1	1	4	2	5	2	15
9	7	2	4	13	2	4	32
10	13	7	2	5	2	4	33
11	2	7	13	13	4	4	43
12	13	2	1	4	13	2	35
13	1	4	2	5	4	5	21
14	13	7	2	5	1	5	33
15	13	7	2	4	1	1	28
16	1	2	1	4	7	5	20
17	7	4	13	7	1	7	39
18	7	5	1	5	4	13	35
19	5	5	13	7	13	7	50
20	2	13	1	13	1	13	43

Table A.104 Bootstrap Samples, Replication 4, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7	4	7	7	4	1	30
2	2	13	4	1	2	5	27
3	1	1	7	4	4	5	22
4	2	4	1	5	5	13	30
5	2	2	2	13	1	4	24
6	13	4	5	7	4	7	40
7	2	13	1	1	7	4	28
8	7	5	1	1	2	13	29
9	13	2	5	5	7	7	39
10	1	4	13	7	13	13	51
11	7	1	1	4	7	5	25
12	5	5	13	5	2	7	37
13	5	2	2	2	5	2	18
14	2	7	7	2	2	4	24
15	1	1	7	13	4	1	27
16	13	1	1	2	2	5	24
17	7	7	5	7	2	7	35
18	2	13	5	5	7	1	33
19	13	4	4	7	7	7	42
20	7	5	5	13	7	4	41

Table A.105 Bootstrap Samples, Replication 5, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7	13	7	1	13	1	42
2	7	13	1	2	7	7	37
3	7	5	1	2	7	7	29
4	7	13	7	1	2	7	37
5	5	4	1	4	4	5	23
6	7	1	7	5	1	4	25
7	1	2	13	13	7	13	49
8	2	5	7	2	4	1	21
9	4	1	4	4	5	4	22
10	13	2	4	7	5	5	36
11	5	7	7	4	2	2	27
12	4	13	7	2	2	13	41
13	7	1	7	13	4	4	36
14	7	4	5	7	7	2	32
15	1	13	7	7	2	4	34
16	13	1	1	2	4	1	22
17	4	2	13	7	5	4	35
18	7	5	4	7	13	5	41
19	1	4	1	5	13	13	37
20	1	13	7	7	7	4	39

Table A.106 Bootstrap Samples, Replication 6, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4	13	13	5	4	7	46
2	1	13	4	13	4	1	36
3	1	5	13	7	7	4	37
4	4	13	5	7	2	5	36
5	5	7	7	7	5	13	44
6	1	1	5	7	13	4	31
7	7	13	5	7	5	4	41
8	1	13	1	13	5	7	40
9	7	1	5	7	7	1	28
10	4	4	2	1	1	2	14
11	1	7	2	1	4	13	28
12	1	2	2	7	7	5	24
13	7	4	4	2	7	4	28
14	13	7	5	2	1	2	30
15	4	13	7	4	7	7	42
16	5	2	1	13	4	4	29
17	13	1	4	4	13	5	40
18	2	13	1	7	4	7	34
19	5	13	5	5	2	7	37
20	7	7	7	2	2	7	32

Table A.107 Bootstrap Samples, Replication 7, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5	13	1	1	7	4	31
2	2	13	1	5	5	7	33
3	13	7	13	13	4	13	63
4	5	1	5	5	13	1	30
5	5	5	7	1	2	2	22
6	5	1	5	13	7	1	32
7	2	1	7	7	7	7	31
8	4	4	7	13	1	5	34
9	13	5	1	7	13	13	52
10	2	4	5	7	2	13	33
11	5	4	1	1	2	1	14
12	5	7	5	7	5	7	36
13	4	5	7	2	13	2	33
14	7	5	1	7	5	2	27
15	13	13	5	13	1	1	46
16	7	1	4	13	7	2	34
17	13	4	5	1	13	5	41
18	5	1	1	2	13	2	24
19	1	2	5	5	5	7	25
20	13	7	13	4	5	1	43

Table A.108 Bootstrap Samples, Replication 8, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5	7	13	4	4	4	37
2	7	7	13	4	13	4	48
3	7	2	7	1	2	2	21
4	2	2	1	1	7	4	17
5	7	13	5	7	1	5	38
6	2	13	2	7	5	4	33
7	13	1	7	7	5	2	35
8	1	7	2	13	2	7	32
9	5	4	7	1	4	2	23
10	7	13	5	7	4	13	49
11	7	4	7	2	1	1	22
12	7	13	5	4	7	7	43
13	4	13	13	5	2	13	50
14	7	4	5	2	7	4	29
15	13	7	1	2	7	7	37
16	2	13	2	1	2	4	24
17	2	1	4	1	4	5	17
18	2	2	5	1	4	5	19
19	13	13	13	2	5	1	47
20	13	4	2	13	1	2	35

Table A.109 Bootstrap Samples, Replication 9, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7	2	7	5	13	13	47
2	2	4	1	13	4	1	25
3	4	7	7	4	2	13	37
4	2	1	13	4	5	5	30
5	4	1	13	13	1	13	45
6	13	7	1	5	4	5	35
7	4	13	5	1	13	2	38
8	5	4	13	2	5	5	34
9	5	13	4	2	4	2	30
10	4	4	2	1	2	1	14
11	13	1	4	5	4	13	40
12	4	5	7	7	4	1	28
13	1	4	7	5	7	7	31
14	1	4	5	1	5	5	21
15	7	2	7	13	1	7	37
16	13	7	2	5	2	4	33
17	1	2	5	7	4	2	21
18	13	2	7	13	7	4	46
19	7	4	13	4	2	1	31
20	7	1	5	7	2	7	29

Table A.110 Bootstrap Samples, Replication 10, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	13	13	7	13	13	7	66
2	2	2	1	4	13	13	35
3	1	1	13	7	1	2	25
4	4	5	7	4	2	5	27
5	7	13	4	7	13	2	46
6	5	13	4	4	13	5	44
7	1	13	7	5	7	2	35
8	1	2	1	7	2	2	15
9	7	5	5	13	2	1	33
10	1	5	2	7	4	4	23
11	13	5	4	4	13	1	40
12	1	1	5	4	13	1	25
13	2	1	1	1	2	13	20
14	1	7	7	13	5	5	38
15	5	7	13	2	1	4	32
16	5	1	5	2	2	7	22
17	7	5	7	2	7	13	41
18	5	7	5	5	2	1	25
19	4	2	1	13	7	1	28
20	13	2	5	5	7	4	36

Table A.111 Bootstrap Samples, Replication 11, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4	2	7	7	4	7	31
2	5	13	1	13	7	1	40
3	1	7	7	13	2	13	43
4	2	1	13	13	7	5	41
5	2	13	7	1	2	13	38
6	7	4	13	1	1	1	27
7	13	2	2	7	4	1	29
8	5	7	4	1	4	1	22
9	13	13	1	13	13	7	60
10	5	2	13	2	13	5	40
11	7	13	4	7	7	13	51
12	7	5	2	1	4	13	32
13	1	13	4	2	5	1	26
14	5	5	7	4	1	5	27
15	5	4	4	1	4	4	22
16	5	13	4	2	5	13	42
17	4	1	13	4	1	13	36
18	1	1	2	4	2	4	14
19	5	7	13	7	1	4	37
20	4	1	13	2	1	1	22

Table A.112 Bootstrap Samples, Replication 12, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5	7	7	1	5	7	32
2	5	2	7	7	5	4	30
3	5	5	2	7	5	1	25
4	13	5	4	5	2	2	31
5	4	2	4	13	4	2	29
6	5	5	13	13	2	1	39
7	2	2	7	1	2	13	27
8	4	5	2	5	13	7	36
9	1	2	5	5	13	4	30
10	1	1	4	7	5	4	22
11	4	5	2	1	7	4	23
12	7	13	5	2	13	2	42
13	2	7	7	4	2	13	35
14	2	4	7	13	1	1	28
15	4	13	2	5	13	4	41
16	7	5	1	4	7	2	26
17	2	7	5	13	7	4	38
18	13	4	2	4	5	4	32
19	4	5	1	4	4	13	31
20	7	7	1	1	4	4	24

Table A.113 Bootstrap Samples, Replication 13, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	13	7	13	5	2	4	44
2	7	13	13	7	2	2	44
3	7	1	7	13	4	4	36
4	1	2	13	4	2	2	24
5	4	1	5	1	4	7	22
6	7	1	7	2	13	1	31
7	2	2	7	7	4	2	24
8	1	7	13	2	2	2	27
9	13	5	5	4	7	5	39
10	13	1	7	7	4	2	34
11	13	1	7	5	5	5	36
12	7	7	5	7	5	7	38
13	2	7	7	5	1	7	29
14	5	1	7	13	5	7	38
15	5	7	4	4	13	7	40
16	4	4	5	1	1	5	20
17	2	7	7	13	4	5	38
18	2	1	13	1	5	5	27
19	5	2	1	7	13	4	32
20	4	5	4	7	13	13	46

Table A.114 Bootstrap Samples, Replication 14, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	13	1	1	1	4	7	27
2	1	5	7	7	2	7	29
3	7	5	2	13	1	4	32
4	4	2	7	2	13	13	41
5	7	4	4	2	2	2	21
6	1	5	5	2	5	1	19
7	2	2	2	5	2	4	17
8	4	2	13	13	13	1	46
9	4	2	4	7	4	13	34
10	7	5	13	7	1	4	37
11	5	1	1	2	7	1	17
12	13	13	4	4	2	2	38
13	5	7	5	4	2	2	25
14	2	5	7	5	4	2	25
15	7	1	2	13	1	7	31
16	2	5	2	5	4	2	20
17	2	5	4	13	4	1	29
18	7	5	13	2	1	7	35
19	1	2	1	7	1	2	14
20	7	13	7	7	7	1	42

Table A.115 Bootstrap Samples, Replication 15, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	2	7	5	13	4	5	36
2	2	7	7	1	4	5	26
3	4	7	2	1	1	13	28
4	7	13	13	4	4	5	46
5	1	13	13	2	4	5	38
6	2	2	2	7	1	13	27
7	7	4	1	4	7	13	36
8	1	7	2	7	7	2	26
9	4	5	7	7	5	13	41
10	2	4	2	4	2	5	19
11	13	13	4	13	1	4	48
12	5	2	2	4	2	13	28
13	2	7	5	4	2	13	33
14	1	5	5	5	13	7	36
15	13	7	2	2	2	2	28
16	13	1	7	5	1	7	34
17	2	2	2	1	5	1	13
18	13	4	5	2	5	4	33
19	13	5	4	2	13	13	50
20	7	1	4	2	1	5	20

Table A.116 Bootstrap Samples, Replication 16, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	1	2	7	5	4	7	26
2	7	2	5	13	1	5	33
3	13	7	13	2	5	13	53
4	13	5	2	7	5	13	45
5	4	7	7	1	13	7	39
6	5	4	7	5	2	4	27
7	5	2	7	4	4	1	23
8	4	1	1	13	4	7	30
9	13	2	2	7	7	4	35
10	1	5	2	13	13	4	38
11	2	2	1	2	13	13	33
12	1	5	2	1	7	4	20
13	4	1	7	2	1	13	28
14	7	1	2	7	1	1	19
15	1	1	2	2	5	4	15
16	1	2	2	4	7	7	23
17	4	13	4	13	5	4	43
18	1	1	5	1	7	4	19
19	5	2	7	7	13	1	35
20	1	4	1	13	13	1	33

Table A.117 Bootstrap Samples, Replication 17, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	7	1	13	7	2	13	43
2	13	1	1	1	4	7	27
3	1	4	2	5	4	7	23
4	2	5	13	13	4	5	42
5	5	5	5	1	2	5	23
6	5	2	7	4	2	2	22
7	4	7	2	2	1	7	23
8	13	7	7	2	13	1	43
9	2	1	13	13	13	1	43
10	13	2	4	7	13	2	41
11	2	2	4	2	4	4	18
12	4	4	7	1	7	1	24
13	7	13	2	13	7	7	49
14	5	1	4	13	13	2	38
15	5	4	2	7	5	5	28
16	2	1	13	1	7	5	29
17	2	7	13	4	4	2	32
18	4	7	4	7	2	7	31
19	5	13	7	1	13	4	43
20	1	5	4	7	2	1	20

Table A.118 Bootstrap Samples, Replication 18, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	4	1	13	7	5	5	35
2	2	1	13	1	2	2	21
3	5	13	1	5	5	13	42
4	4	13	7	1	4	4	33
5	2	13	1	5	13	4	38
6	13	7	1	5	4	7	37
7	4	2	5	7	2	5	25
8	5	4	5	1	5	2	22
9	7	1	2	7	13	1	31
10	2	2	13	2	2	4	25
11	4	1	1	13	7	13	39
12	1	5	4	2	4	13	29
13	7	2	1	4	4	1	19
14	7	1	4	7	2	7	28
15	4	5	1	5	4	13	32
16	1	1	1	5	7	4	19
17	7	1	7	13	4	1	33
18	1	2	7	4	7	1	22
19	4	5	2	4	5	13	33
20	13	5	1	13	7	1	40

Table A.119 Bootstrap Samples, Replication 19, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	13	7	5	13	5	5	48
2	4	7	5	7	13	2	38
3	4	5	7	1	13	2	32
4	1	7	13	2	2	1	26
5	5	4	7	4	5	13	38
6	13	2	13	7	7	1	43
7	7	2	5	1	7	13	35
8	4	2	1	1	5	13	26
9	4	1	5	13	13	13	49
10	5	13	4	4	2	5	33
11	5	13	7	5	13	4	47
12	13	5	5	1	4	5	33
13	1	1	1	5	1	2	11
14	1	5	4	5	5	2	22
15	4	2	13	13	5	2	39
16	1	7	4	5	7	1	25
17	13	5	1	4	13	13	49
18	2	1	5	7	2	5	22
19	13	1	13	4	7	2	40
20	7	1	4	5	7	7	31

Table A.120 Bootstrap Samples, Replication 20, Scenario 2 U-Boat Kills

Trial	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Total
1	5	2	7	2	5	4	25
2	4	7	1	13	7	1	33
3	4	7	5	7	1	4	28
4	1	5	1	5	5	2	19
5	5	7	4	4	13	4	37
6	2	5	5	13	4	13	42
7	4	4	13	1	7	2	31
8	4	13	13	4	7	5	46
9	2	1	13	2	1	5	24
10	2	13	2	5	7	5	34
11	5	5	4	5	7	1	27
12	4	2	13	2	5	5	31
13	1	5	2	5	1	1	15
14	7	1	7	13	4	4	36
15	5	1	7	1	1	7	22
16	2	5	1	7	13	1	29
17	2	13	1	4	1	13	34
18	1	7	2	13	5	7	35
19	5	4	7	13	7	4	40
20	13	5	5	2	5	7	37

Appendix B. Model Implementation

Appendix B contains specific details about the implementation of the Bay of Biscay simulation in the agent-based paradigm. Figure B.1 shows the Java inheritance of the major component classes.

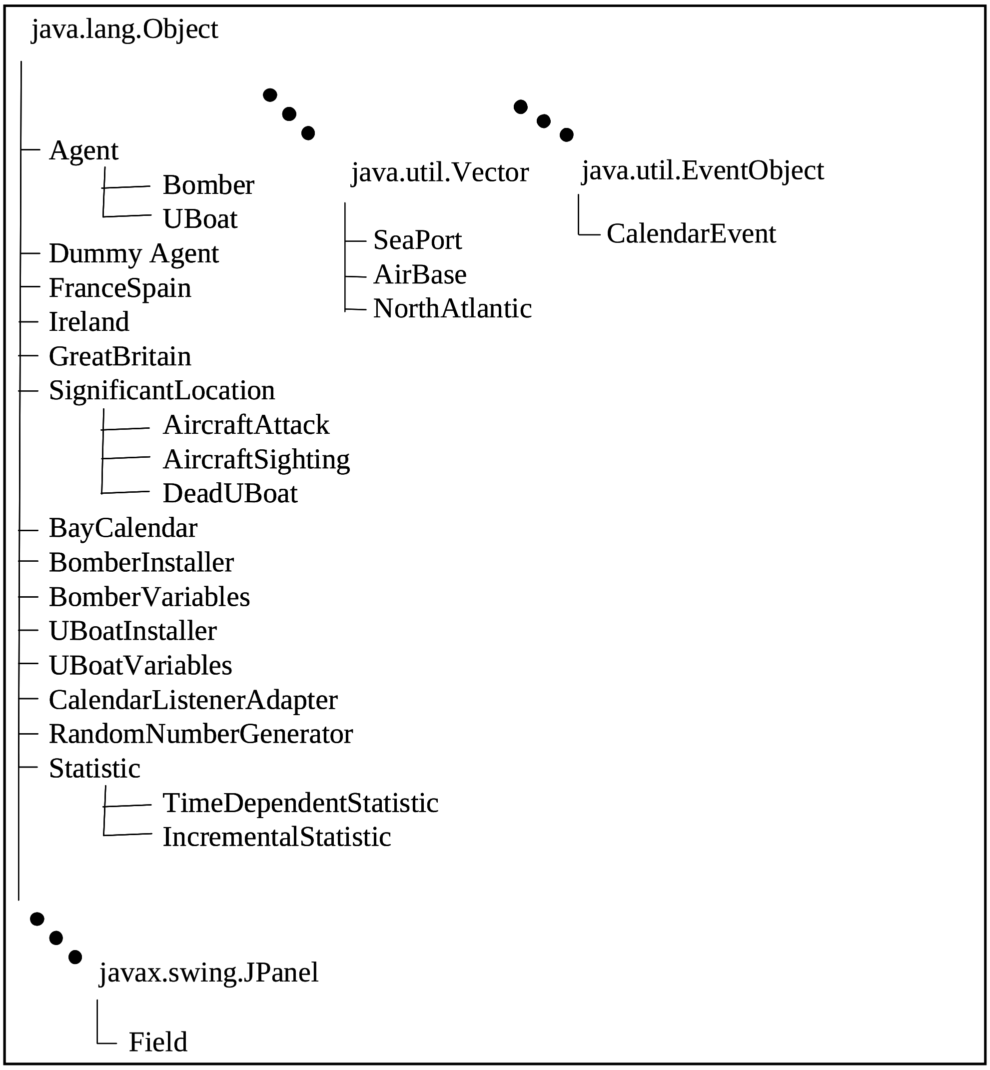

Figure B.1 Simulation Class Inheritance Diagram

The remaining sections of Appendix B illustrate the implementation of specific portions of the Bay of Biscay agent-based simulation. The flow diagrams are intended to

augment the discussions of the simulation implementation within this document and aid follow-on research efforts that attempt to recreate the results presented.

B.1 Aircraft Agent Algorithms

Figures B.2 – B.4 present the majority of the algorithms responsible for the aircraft agents' decisions and actions. Implementing the Runnable interface, aircraft agent code overrides the run method to provide its individual thread with instructions. Figure B.2 details the run method, which requests permission to act from the simulation clock manager (the Field object). Except for checking for maintenance cancellations if the agent is located at the airbase, the run method passes control to the update method for aircraft activity.

The most notable aspect of the run method occurs when the aircraft agent requests permission to act before the simulation clock has reached the agent's scheduled action time. In this case, the Field object, which controls the simulation clock, puts the agent thread to sleep. This is an essential aspect of the simulation because it prevents the agent from repeatedly attempting to act, thereby monopolizing the CPU and preventing other agents from acting. When the Field object advances the simulation clock, the sleeping agents are notified, and they can request permission to act again.

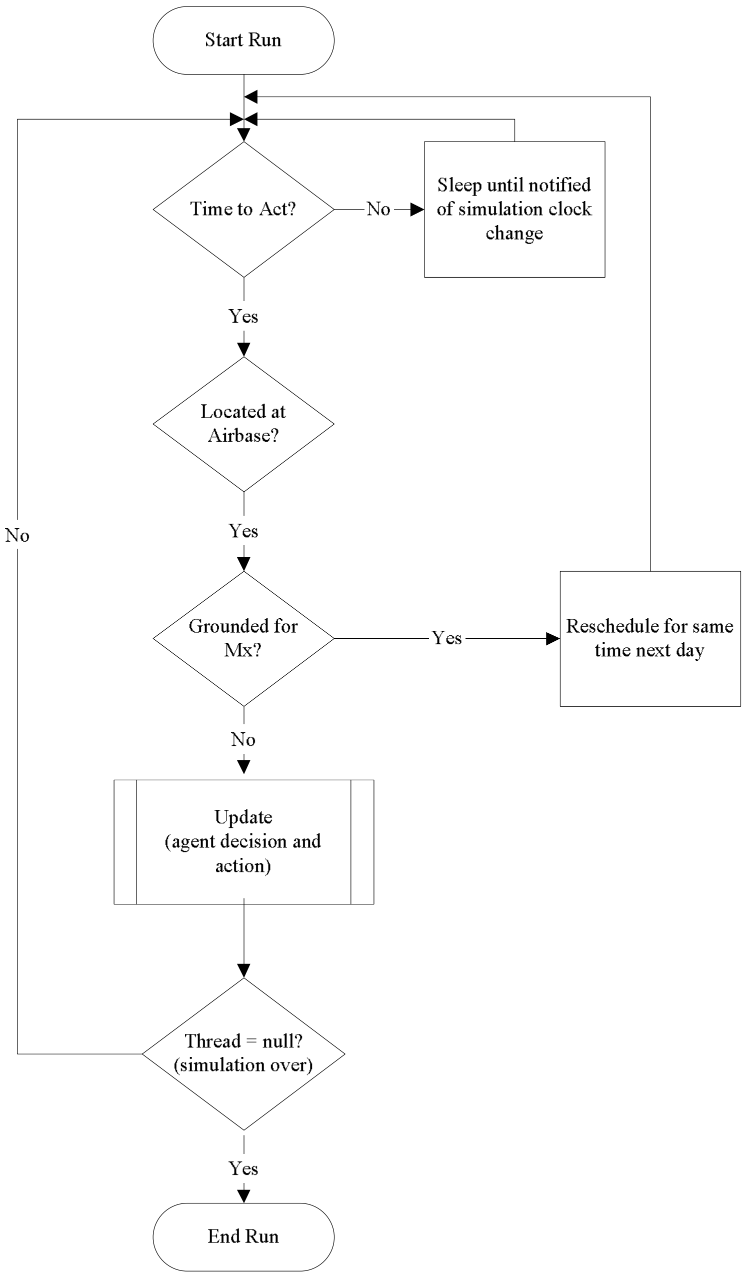

Figure B.2 Bomber Agent Run Method Algorithm

Figure B.3 details the aircraft agent's update method. The activities and decisions represented in Figure B.3 were sufficiently detailed in the text of this document. However, the flow diagram shows the precedence of the various decisions and actions.

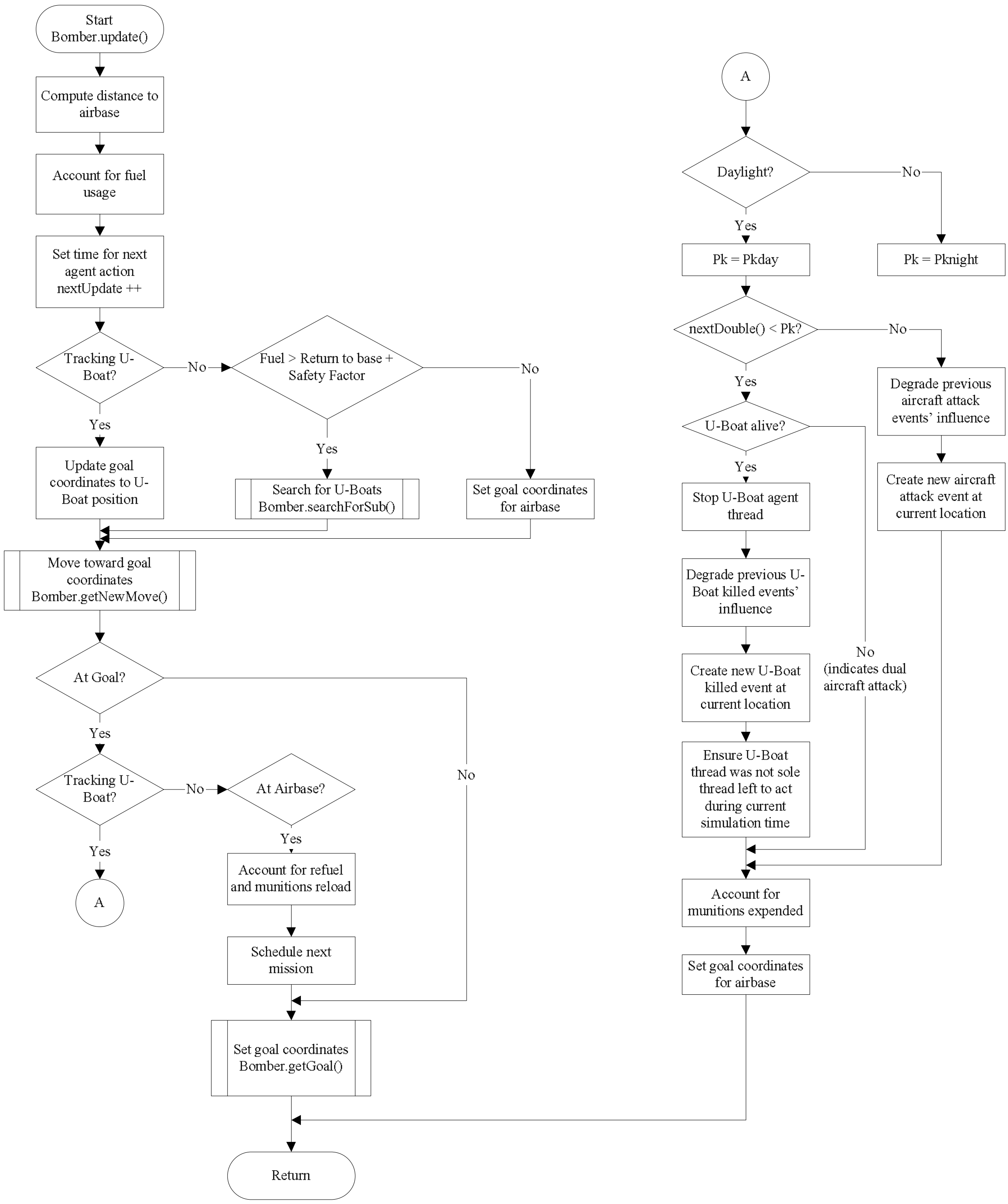

Figure B.3 Bomber Agent Update Method Algorithm

Finally, Figure B.4 details the method used to determine whether or not an aircraft agent detects a U-Boat within its effective search range. The aircraft checks its range to

each U-Boat in the simulation to determine whether or not it is within the combined sensor sweep width. If the U-Boat is outside the sweep width, then the aircraft checks the next U-Boat. However, if the U-Boat is within the sweep width, then the aircraft makes a random draw against the computed probability of detection [McCue, 1990]. If a U-Boat is detected, then the aircraft immediately stops searching for others that may be in the area. Therefore, only the location of the first U-Boat detected by an aircraft will be discovered on any given aircraft sortie.

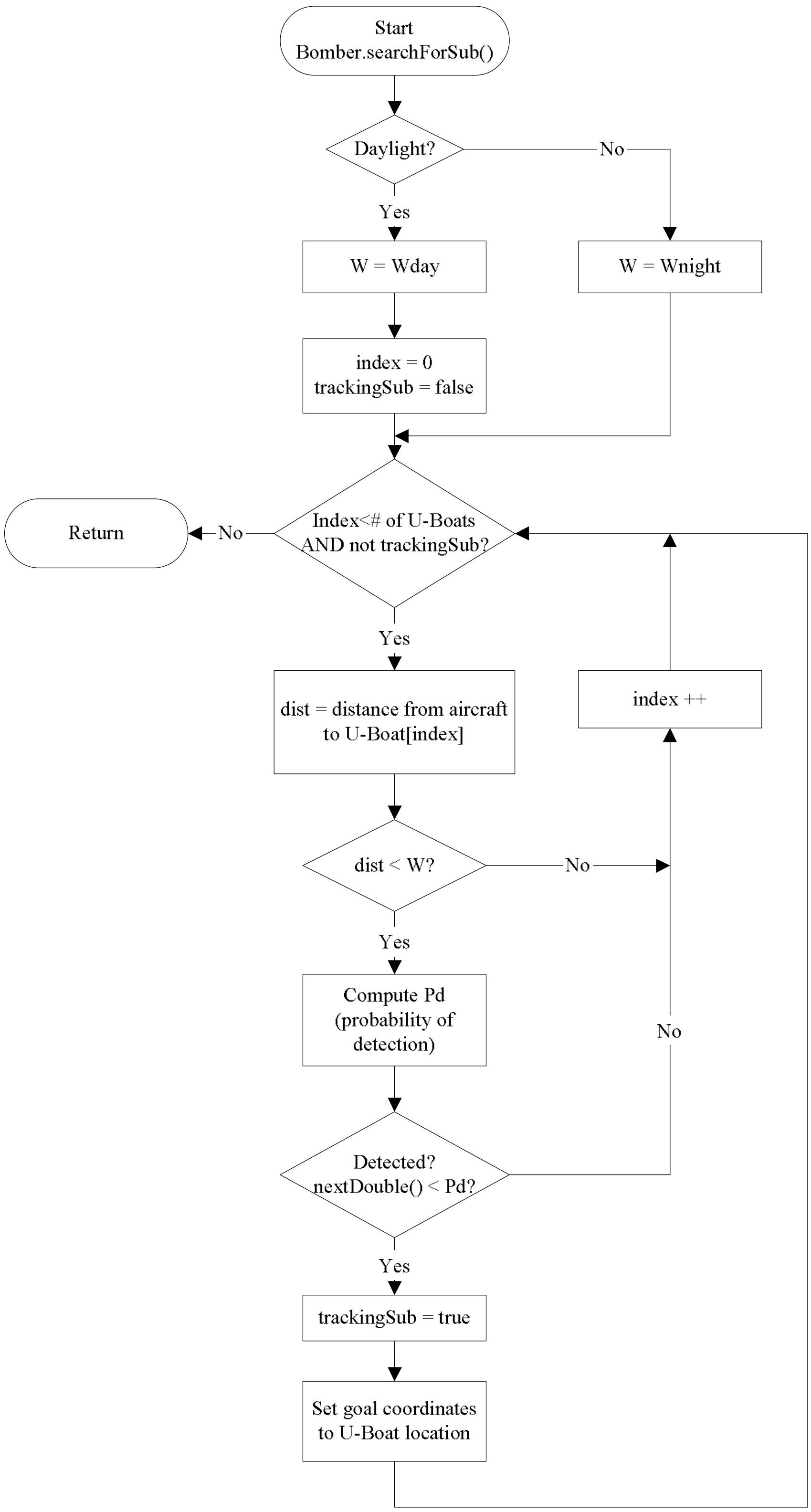

Figure B.4 Bomber Agent U-Boat Detection Algorithm

B.2 U-Boat Agent Algorithms

Figures B.5 – B.6 present the majority of the algorithms responsible for the U-Boat agents' decisions and actions. Implementing the Runnable interface, U-Boat agent code overrides the run method to provide its individual thread with instructions. Figure B.5 details the run method, which requests permission to act from the simulation clock manager (the Field object). Though the update method in Figure B.6 contains the majority of the agent decision/action code, the run method has the job of setting the goal coordinates of U-Boat agents when entering the Bay of Biscay from either operations in the North Atlantic or port.

Like the aircraft agent run method, a U-Boat agent requesting permission to act at a time later than the current simulation clock value is put to sleep. When the Field object advances the simulation clock, the sleeping agents are notified, and they can request permission to act again.

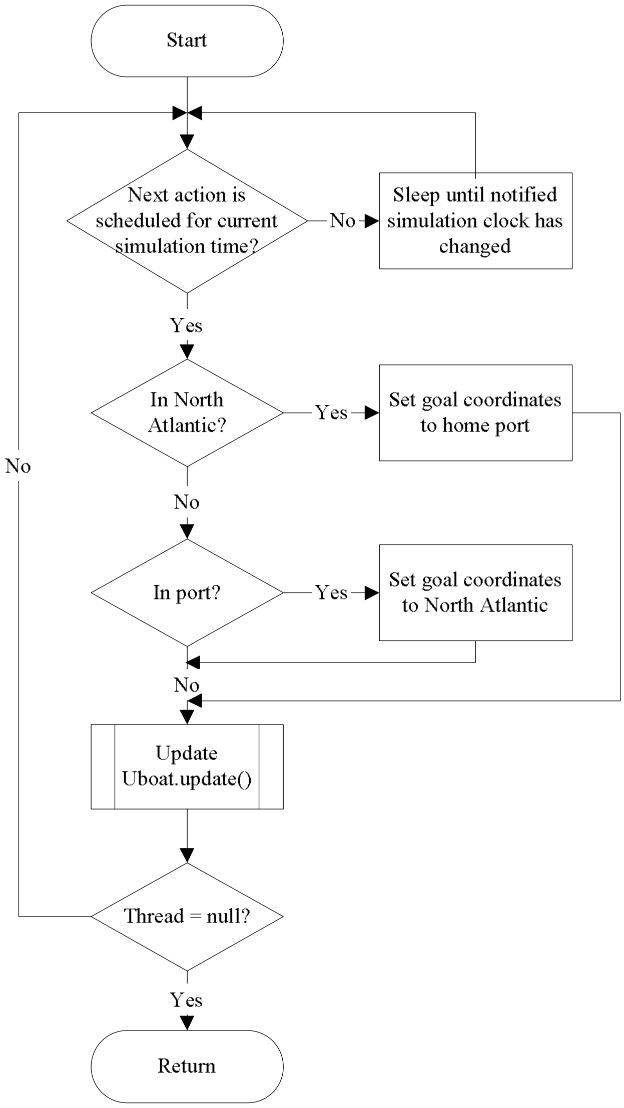

Figure B.5 U-Boat Agent Run Method Algorithm

Figure B.6 details a U-Boat agent's update method. The activities and decisions represented in Figure B.6 were sufficiently detailed in the text of this document. However, the flow diagram shows the precedence of the various decisions and actions.

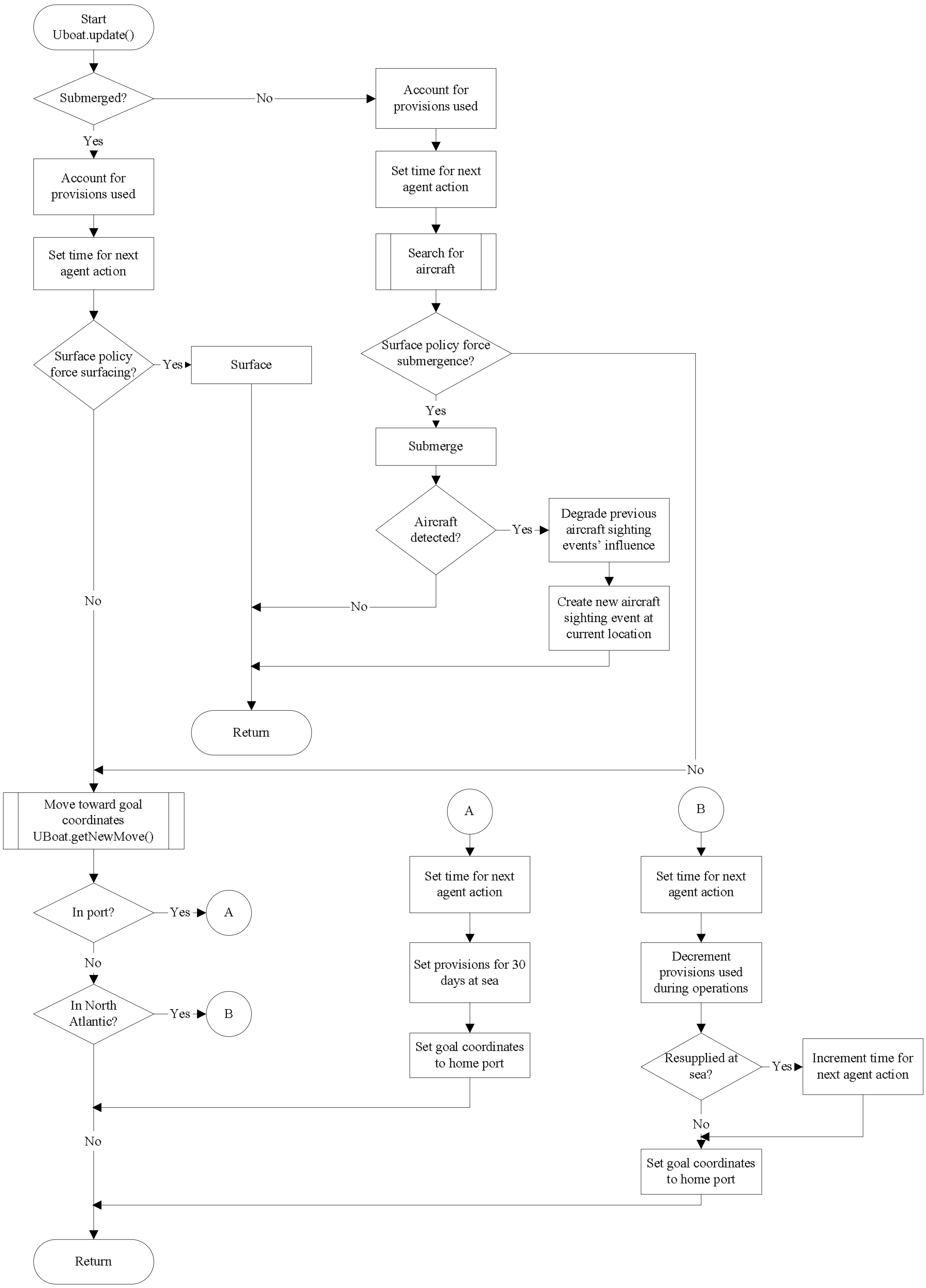

Figure B.6 U-Boat Agent Update Method Algorithm

B.3 Simulation Environment

The Field class was the simulation environment in which the system agents were situated. The agents within the simulated system used a coordinate system relative to the Field object's JPanel coordinates. Classes representing the landmasses surrounding and defining the Bay of Biscay – Ireland, GreatBritain, and FranceSpain (Figure B.1) – further define the agents' environment. Additionally, the Field object maintained the system clock and served as a broker for the agents wanting to act. It is this function that is shown in the flow diagram of Figure B.7.

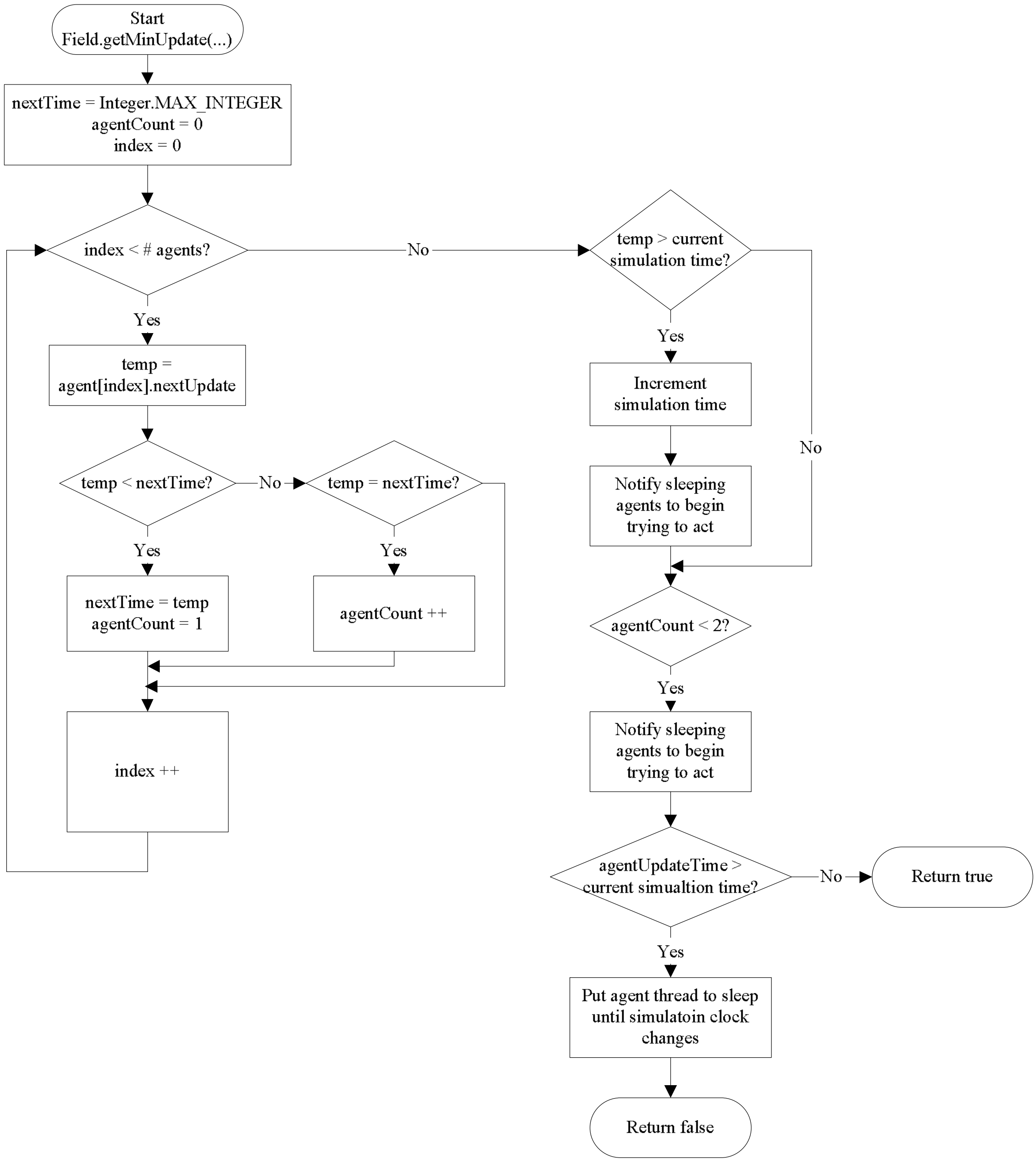

Figure B.7 Field getMinUpdate Method Algorithm Used to Advance the Simulation Clock and Control Agent Timing

When requesting permission to act, the Field object makes several calculations. First, it determines the nearest (in the sense of future) time any agent is scheduled to act and the number of agents that are scheduled for that time. If the nearest time is later than

the current simulation clock time, then the simulation clock is advanced and all sleeping agents are notified (awoken). If not, the requesting agent's schedule is compared to the simulation clock. If the scheduled time is later, the thread is told to sleep (and returns false); otherwise, the agent is given permission to act (i.e. returns true). This system prevents an agent from acting prior to its scheduled time and also allows for a single request during any simulation time increment (in practice speeding up simulation run time significantly).

References

1. 2nd Euromicro Workshop on Component-based Software website. 28th Euromicro Conference. http://www.idt.mdh.se/ecbse/2002/

2. Aeronautical Systems Center (ASC/XREWS) briefing (1992) "Theater Attack Model". Eglin Air Force Base FL. UNCLASSIFIED.

3. Air Force Doctrine Document 2 (AFDD-2). HQ AFDC/DR. 17 February 2000.

4. Air Force Studies and Analyses Agency (AFSAA) briefing (2000) "Combined Forces Assessment Model (CFAM)". Washington DC. UNCLASSIFIED.

5. Aldridge, Rich (editor) (2000) *Course 26C: International Studies*. Squadron Officer College Distributive Learning. Maxwell Air Force Base AL. June 2000.

6. Archambeault, Bruce (1999) "Modeling and Simulation Validation for EMC Applications". *1999 IEEE International Symposium on Electromagnetic Compatibility*, vol. 1. pp. 492- 496.

7. Arthur, James D. and Richard E. Nance (2000) "Verification and Validation without Independence: A Recipe for Failure". *Proceedings of the 2000 Winter Simulation Conference*. eds. J. A. Joines, R. E. Barton, K. Kang, and P. A. Fishwick. pp. 859-865.

8. Axelrod, Robert and Michael Cohen (2000) *Harnessing Complexity: Organizational Implications of a Scientific Frontier*. The Free Press, a division of Simon & Schuster Inc.: New York NY.

9. Balci, Osman (1994) "Validation, Verification, and Testing Throughout the Life Cycle of a Simulation Study". *Annals of Operation Research*, vol. 54. pp. 121-174.

10. Balci, Osman (1995) "Principles and Techniques of Simulation Validation, Verification, and Testing". *Proceedings of the 1995 Winter Simulation Conference*. eds. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman. pp. 147-154.

11. Balci, Osman (2001) "A Methodology for Certification of Modeling and Simulation Applications". *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 11, no. 4. October.

12. Balci, Osman and Robert G. Sargent (1982) "Validation of Multivariate Response Models Using Hotelling's Two-Sample T^2 Test". *Simulation*. December. pp. 185-192.

13. Balci, Osman and Robert G. Sargent (1984) “Validation of Simulation Models Via Simultaneous Confidence Intervals”. *American Journal of Mathematical and Management Sciences*, vol. 4, nos. 3 & 4. pp. 375-406.

14. Balci, Osman and William F. Ormsby (2002) “Expanding Our Horizons in Verification, Validation, and Accreditation Research and Practice”. *Proceedings of the 2002 Winter Simulation Conference*. eds. E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes. December.

15. Banks, Jerry, John S. Carson, and Barry L. Nelson (1996) *Discrete-Event System Simulation (2nd ed.)*. Prentice-Hall, Inc.: Upper Saddle River NJ. pp. 406-424.

16. Bauer, Kenneth. OPER 685 – Multivariate Analysis Class Notes. Air Force Institute of Technology: Wright-Patterson AFB OH.

17. Bauer, Kenneth. OPER 785 – Advanced Multivariate Analysis Class Notes. Air Force Institute of Technology: Wright-Patterson AFB OH.

18. Beckerman, Linda. Is the Time for Revolution Upon Us? http://www.sisostds.org/webletter/siso/iss_51/art_236.htm

19. Behnia, Masud, Wataru Nakayama, and Jeffrey Wang (1998) “CFD Simulations of Heat Transfer from a Heated Module in an Air Stream: Comparison with Experiments and a Parametric Study”. *The Sixth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, 1998*. pp. 143-151.

20. Bergeman, Russell (2001) “Considering the Intangibles: Identifying Social Capital in Military Units,” *Maneuver Warfare Science 2001*, Gary Horne and Mary Leonardi, eds., Defense Automated Printing Service: Quantico VA.

21. Bergeson, Pam (editor) (2000) *Course 26B: Military Studies*. Squadron Officer College Distributive Learning. Maxwell Air Force Base AL. June 2000.

22. Berry, Michael (1991) “Quantum Physics on the Edge of Chaos”. *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

23. Boccara, N., O. Roblin, and M. Roger (1994) “Automata Network Predator-Prey Model with Pursuit and Evasion,” *Physical Review E*, vol. 50, no. 6, pp. 4531-4541.

24. Bonabeau, Eric (2002) “Predicting the Unpredictable”. *Harvard Business Review*. vol. 80:3. pp. 109-116. March.

25. Bortfeldt, Andreas and Hermann Gehring (2001) “A Hybrid Genetic Algorithm for the Container Loading Problem”. *European Journal of Operational Research*. vol. 131. pp. 143-161.

26. Botee, Hozefa and Eric Bonabeau (1998) “Evolving Ant Colony Optimization”. *Advanced Complex Systems*. vol. 1. pp. 149-159.

27. Brade, Dirk (2000) “Enhancing Modeling and Simulation Accreditation by Structuring Verification and Validation Results”. *Proceedings of the 2000 Winter Simulation Conference*. eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick. pp. 840-848.

28. Brandstein, Alfred and Gary Horne (1988) “Data Farming: A Meta-technique for Research in the 21st Century”. *Maneuver Warfare Science 1988*. edited by F. G. Hoffman and Gary Horne. United States Marine Corps.

29. Brown, Lloyd (2000) “Agent-based Simulation as an Exploratory Tool in the Study of the Human Dimension of Combat,” Naval Postgraduate School: Monterey CA.

30. Bullock, Kelly (2000) “Hierarchical Interactive Theater Model (HITM),” Air Force Institute of Technology: WPAFB OH. AFIT/GOA/ENS/00M-05.

31. Cameron, Stewart, Guillermo Loubriel, Rush Robinett III, Keith Stantz, Michael Trahan, and John Wagner (1999) “Adaptive Remote-Sensing Techniques Implementing Swarms of Mobile Agents”. *SPIE Conference on Unattended Ground Sensor Technologies and Applications*. SPIE vol. 3714. April 1999.

32. Carl, R. Greg (2003) “Search Theory and U-Boats in the Bay of Biscay”. *OR/MS Tomorrow*, vol. V. Spring 2003.

33. Caughlin, Don (2000) “An Integrated Approach to Verification, Validation, and Accreditation of Models and Simulations”. *Proceedings of the 2000 Winter Simulation Conference*. eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick. pp. 872-881.

34. Ceranowicz, Andy (1994) Modular Semi-Automated Forces. Society for Computer Simulation International.

35. Champagne, Lance (2000) “Modeling in the Air Force”. Briefing to OPER 400 class. 21 September. UNCLASSIFIED.

36. Champagne, Lance (2001a) “Chaos, Complexity, and Artificial Life”. Air Force Institute of Technology Technical Report. AFIT/ENS-TR-02-01.

37. Champagne, Lance (2001b) “Agent-Based Optimization”. Air Force Institute of Technology Technical Report. AFIT/ENS-TR-02-02.

38. Champagne, Lance (2001c) “Human and Organizational Behavior Modeling”. Air Force Institute of Technology Technical Report. AFIT/ENS-TR-02-03.

39. Champagne, Lance and Ray Hill (2003) “Multi-Agent Simulation Analysis: Bay of Biscay Case Study”. *Proceedings of SimTecT 2003*. Adelaide, Australia. May 26-29.

40. Champagne, Lance, R. Greg Carl, and Ray Hill (2003a) “Multi-Agent Techniques: Hunting U-Boats in the Bay of Biscay”. *Proceedings of SimTecT 2003*. Adelaide, Australia. May 26-29.

41. Champagne, L., R. G. Carl, and R. Hill (2003b) “Search Theory, Agent-Based Simulation, and U-Boats in the Bay of Biscay” *Proceedings of the 2003 Winter Simulation Conference*. S. Chick, P.J. Sanchez, D. Ferrin, and D. J. Morrice, eds., IEEE. New Orleans LA. [To appear]

42. Champagne, L. (2003) “Bay of Biscay: Extensions into Modern Military Issues”. *Proceedings of the 2003 Winter Simulation Conference*. S. Chick, P.J. Sanchez, D. Ferrin, and D. J. Morrice, eds., IEEE. New Orleans LA. [To appear]

43. Cheng, Russell (2001) “Analysis of Simulation Experiments by Bootstrap Resampling”. *Proceedings of the 2001 Winter Simulation Conference*. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.. December.

44. Clerc, Maurice (2000) “Discrete Particle Swarm Optimization: Illustrated by the Traveling Salesman Problem”. http://www.mauriceclerc.net. 29 February 2000.

45. Colorni, A., M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, and M. Trubian “Heuristics from Nature for Hard Combinatorial Optimization Problems”. *International Transactions in Operational Research*. vol. 3, no. 1. pp. 1-21.

46. Colorni, A., M. Dorigo, and V. Maniezzo (1992). “Distributed Optimization by Ant Colonies”. *Proceedings of the First European Conference on Artificial Life*. F. Varela and P. Bourgine, eds. Elsevier Publishing: Paris, France. pp. 134-142.

47. Colorni, A., M. Dorigo, and V. Maniezzo (1992). “An Investigation of Some Properties of an Ant Algorithm”. *Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92)*. R. Manner and B. Manderick, eds. Elsevier Publishing: Brussels, Belgium. pp. 509-520.

48. Connor, George (1997) “A Commentary”. *Airpower Journal*, vol. XI, no. 1. Air University: Montgomery AL. Spring 1997. pp. 97-98.

49. Conover, W. J. (1999) *Practical Nonparametric Statistics, Third Edition*. John Wiley and Sons, Inc.: New York NY.

50. Corne, David, Marco Dorigo, and Fred Glover (1999) “Introduction”. *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England. 1999.

51. Dasgupta, Dipankar (1999) "Information Processing in the Immune System". *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England.

52. Davis, Paul K. (1992) Generalized Concepts and Methods of Verification, Validation, and Accreditation (VV&A) for Military Simulations. RAND: Santa Monica CA. R-4249-ACQ.

53. Deitel, H. M. and P. J. Deitel (2002) *JAVA™: How to Program*. Prentice Hall, Inc.: Upper Saddle River NJ.

54. Department of Defense Instruction 5000.61 (1996) "DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)". April 29, 1996.

55. Department of Defense Instruction 5000.61, Draft (2002) "DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)". July 10, 2002.

56. Dillon, William and Matthew Goldstein. *Multivariate Analysis: Methods and Applications*. John Wiley & Sons, Inc.: New York. 1984.

57. Dixon, Philip M. (2001) "The Bootstrap". Department of Statistics, Iowa State University. 20 December.

58. Dorigo, Marco and Gianni Di Caro, (1999) "The Particle Swarm: Social Adaptation in Information-Processing Systems". *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England. 1999.

59. Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni (1996) "The Ant System: Optimization by a Colony of Cooperating Agents". *IEEE Transactions on Systems, Man, and Cybernetics – Part B*. vol. 26, no. 1. pp. 1-14.

60. Dunnigan, James (2000) *Wargames Handbook, Third Edition: How to Play and Design Commercial and Professional Wargames*. Writers Club Press: San Jose CA.

61. Efron, B. (1979) "Bootstrap Methods: Another Look at the Jacknife". *Annals of Statistics*, vol. 7. pp. 1-26.

62. Efron, B. and R. Tibshirani (1986) "Bootstrap Methods for Standard Errors, Confidence Intervals and Other Measures of Statistical Accuracy". *Statistical Science*, vol. 1. pp. 54-77.

63. Erlenbruch, Thomas (2002) "Agent-Based Simulation of German Peacekeeping Operations for Units up to Platoon Level," Naval Postgraduate School: Monterey, CA.

64. Ferber, Jaques. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, an imprint of Pearson Education Ltd.: Harlow, England. 1999.

65. Fordham website. http://www.fordham.edu/halsall/ancient/polybuis-cannae.htm

66. Forrester, J. W., and P. M. Senge (1980) "Tests for Building Confidence in System Dynamics Models". *HMS Studies in the Management Sciences*, vol. 14. pp. 209-228.

67. Fraedrich, D. and A. Goldberg (2000) "A Methodological Framework for the Validation of Predictive Simulations". *European Journal of Operational Research*, vol. 124. pp. 55-62.

68. Freisleben, Bernd and Peter Merz (1996) "A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems". *Proceedings of the IEEE International Conference on Evolutionary Computation IEEE-EC 96*. IEEE Press. pp. 616-621.

69. Gambardella, Luca, Eric Taillard, and Giovanni Agazzi (1999) "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows". *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England. 1999.

70. Gambardella, Luca and Marco Dorigo (1996) "Solving Symmetric and Asymmetric TSPs by Ant Colonies". IEEE 1996.

71. Gaupp, Martin (1999) "Pilot Inventory Complex Adaptive System: An Artificial Life Approach to Managing Pilot Retention". Master's Thesis. Air Force Institute of Technology. AFIT/GOR/ENS/99M-06.

72. Giancoli, Douglas C. (1984) *General Physics*. Prentice-Hall, Inc.: Englewood Cliffs NJ.

73. Gleick, James (1987) *Chaos: Making a New Science*. Penguin Books: New York NY.

74. Glover, Fred and Harvey J. Greenberg (1989) "New approaches for heuristic search: A bilateral linkage with artificial intelligence". *European Journal of Operational Research, vol. 39*. Elsevier Science Publishers: North-Holland. pp. 119-130.

75. Guadiano, Paulo, Benjamin Shargel, Eric Bonabeau, and Bruce T. Clough (2003) "Control of UAV Swarms: What the Bugs Can Teach Us". *Proceedings of the 2nd AIAA Unmanned Unlimited Systems, Technologies, and Operations – Aerospace, Land, and Sea Conference and Workshop*. San Diego CA. September.

76. Gusella, Riccardo (1991) “Characterizing the Variability of Arrival Processes with Indexes of Dispersion”. *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 2. February.

77. Hajela, Prabhat and Jun Sun Yoo (1999) “Immune Network [Modeling] in Design Optimization”. *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England. 1999.

78. Hart, Emma and Ross, Peter (1999) “The Evolution and Analysis of a Potential Antibody Library for Use in Job-Shop Scheduling”. *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England. 1999.

79. Heidemann, John, Mills, Kevin, and Kumar, Sri (2001) “Expanding Confidence in Network Simulations”. *IEEE Network*. September/October. pp. 58-64.

80. Hendler, James (1999) “Guest Editor’s Introduction: Making Sense out of Agents”. *IEEE Intelligent Systems and Their Applications*. vol. 14, no. 2. March/April. pp. 32-37.

81. Hessler, Gunther (1989) *German Naval History: The U-Boat War in the Atlantic, 1939-1945*. Her Majesty’s Stationary Office: London, England.

82. Hill, Raymond, Joseph Price, and Lance Champagne (2003a) “Agent Modeling with Game Theory Constructs”. Summer Simulation Conference, 2003. Montreal, Canada. July.

83. Hill, R. R., G. A. McIntyre, T. R. Tighe, R. K. Bullock (2003b) “Some Experiments with Agent-Based Combat Models. *Military Operations Research*, Vol. 8, No. 3, September.

84. Hodges, James S. (1991) “Six (or so) Things You Can Do with a Bad Model”. *Operations Research*, vol. 39, no. 4. May-June 1991.

85. Hodges, James S. and James A. Dewar (1992) “Is It You or Your Model Talking? A Framework for Model Validation”. Santa Monica CA: RAND. Report R-4114-AF/A/OSD.

86. Holland, John H. (1995) *Hidden Order: How Adaptation Builds Complexity*. Helix Books. Addison-Wesley Publishing Company, Inc.: Reading MA.

87. Holstein, D. and Pablo Moscato (1999) “Memetic Algorithms Using Guided Local Search”. *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England.

88. Horne, Gary E. (2001) Beyond Point Estimates: Operational Synthesis and Data Farming. *Maneuver Warfare Science 2001*. Gary Horne and Mary Leonardi, eds. Defense Automated Printing Service: Quantico VA.

89. Horne, Gary and Mary Leonardi (1998) "Trust on the Battlefield: Exploring Questions with a New Tool". *Maneuver Warfare Science 1998*. F.G. Hoffman and Gary Horne, eds. United States Marine Corps.

90. Hoske, Michael (1999) "Distributed Intelligence: Pouring Thought into the Process". *Control Engineering*. vol. 46, no. 10. October 1999. pp. 48-52.

91. Huang, J. H. (1993) *Sun Tzu: The Art of War. The New Translation*. William Morrow: New York.

92. Ilachinski, Andrew (1998) "Irreducible Semi-Autonomous Adaptive Combat (ISAAC)," *Maneuver Warfare Science 1998,* F.G. Hoffman & Gary Horne, eds., United States Marine Corps.

93. Ilachinski, Andy (2000) Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial Life Approach to Land Combat," *Military Operations Research*, vol. 5, no. 3, pp. 29-46.

94. Jabbar, Shahid and Abbas K. Zaidi (2001) "A Generalized Methodology and Framework for the Validation and Verification of Multi-Agent Systems". *2001 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2. pp. 835-840.

95. James, Glenn (1996) *Chaos Theory – The Essentials for Military Applications*. Naval War College: Newport RI. Paper no. Ten. October.

96. *James Gleick's Chaos: The Software. User Guide*. Autodesk, Inc. Publication CH1RM-02-01. May 14, 1991.

97. Jenkins, Roger, Yogesh Deshpande, and Graydon Davison (1998) "Verification and Validation and Complex Environments: A Study in Service Sector". *Proceedings of the 1998 Winter Simulation Conference*. eds. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan. pp. 1433-1440.

98. Jennings, Nicholas R., Katia Sycara, and Michael Wooldridge (1998) "A Roadmap of Agent Research and Development". *Autonomous Agents and Multi-Agent Systems*, vol 1. pp. 7-38.

99. Jones, Randolph, John Laird, Paul Nielson, Karen Coulter, Patrick Kenney, and Frank Voss (1999) "Automated Intelligent Pilots for Combat Flight Simulation". *AI Magazine*. vol. 20, no. 1. Spring 1999.

100. Kennedy, James and Russell Eberhart (1995) "Particle Swarm Optimization". *Proceedings of the IEEE International Conference on Neural Networks*. Perth, Australia. November 27 – December 1, 1995. pp. 1942-1948.

101. Kennedy, James and Russell Eberhart (1999) "The Particle Swarm: Social Adaptation in Information-Processing Systems". *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England. 1999.

102. Kleijnen, Jack P. C. (1996) "Validation of Trace-Driven Simulation Models: Regression Analysis Revisited". *Proceedings of the 1996 Winter Simulation Conference*. eds. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. pp. 352-359.

103. Kleijnen, Jack P. C. (1995a) "Verification and Validation of Simulation Models". *European Journal of Operational Research*, vol. 82. pp. 145-162.

104. Kleijnen, Jack P. C. (1995b) "Statistical Validation of Simulation Models". *European Journal of Operational Research*, vol. 87. pp. 21-34.

105. Klimack, William K. (2002) "Robustness of Multiple Objective Decision Analysis Preference Functions". Ph.D. Dissertation. Air Force Institute of Technology: Wright-Patterson Air Force Base OH. AFIT/DS/ENS/02-01.

106. Knowles, Joshua and David Corne (2000) "M-PAES: A Memetic Algorithm for Multiobjective Optimization". *Proceedings of the 2000 Congress on Evolutionary Computation*.

107. Knowles, Joshua and David Corne (2000) "A Comparison of Diverse Approaches to Memetic Multiobjective Combinatorial Optimization". Submitted to WOMA: The Workshop on Memetic Algorithms at the 2000 Genetic and Evolutionary Computation Conference.

108. Koopman, Bernard (1970) "A Study of the Logical Basis of Combat Simulation," *Operations Research,* vol. 18, pp. 855-882.

109. Krasnogor, Natalio and Jim Smith (2000) "A Memetic Algorithm with Self-Adaptive Local Search: TSP as a Case Study". *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*.

110. Lauren, Michael (2002) "A Fractal Approach to Equations of Attrition". *Military Operations Research,* 7:3. pp. 17-30.

111. Lauren, Michael (2001) "Applications of a Distillation to Questions for the New Zealand Army," *Maneuver Warfare Science 2001,* Gary Horne and Mary Leonardi, eds., Defense Automated Printing Service: Quantico VA.

112. Lauren, Michael K. (2001) "Complexity Theory and Land Warfare". Defence Technology Agency Briefing. New Zealand Army.

113. Law, Averill and W. David Kelton (1991) *Simulation Modeling and Analysis: Second Edition*. McGraw Hill Inc.: New York.

114. Leingwell, John W. R. (1987) “The Laws of Combat: Lanchester Reexamined”. International Security, vol. 12, no. 1. Summer 1987. pp. 89-134.

115. Lesurf, Jim (1991) “Chaos on the Circuit Board”. *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

116. Levy, Steven (1992) *Artificial Life: A Report from the Frontier Where Computers Meet Biology*, Vintage Books, a division of Random House, Inc.: New York NY.

117. Liu, Bing, Siew-Hwee Choo, Shee-Ling Lok, Sing-Meng Leong, Soo-Chee Lee, Foong-Ping Poon, and Hwee-Har Tan (1994) “Finding the Shortest Route Using Cases, Knowledge, and Dijkstra’s Algorithm”. *AI at Work (CAIA ’94).* IEEE Expert.

118. Looney, Carl. *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists*. Oxford University Press: New York NY. 1997.

119. Lorenz, Edward (1993) *The Essence of Chaos*. University of Washington Press: Seattle WA.

120. Mandelbrot, Benoit (1991) “Fractals – a geometry of nature”. *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

121. May, Robert (1991) “The Chaotic Rhythms of Life”. *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

122. McCue, Brian (1990) *U-Boats in the Bay of Biscay: an essay in Operations Research*. National Defense University Press: Washington DC.

123. McCue, Brian (2002) Interview at Center for Naval Analysis (CNA) discussing modeling issues with respect to the offensive search for U-Boats in the Bay of Biscay during WW II. 23 August 2002, 0900 – 1100 hrs.

124. McDonald, J. Todd and Michael Talbert (2000) Agent-based Architecture for Modeling and Simulation Integration. *Proceedings of the National Aerospace & Electronics Conference (NAECON 2000)*. Dayton OH. 10-12 October 2000.

125. McRobie, Allan and Michael Thompson (1991) “Chaos, catastrophes and engineering”. *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

126. Metz, Michael L. (2000) "Joint Warfare System (JWARS) Verification and Validation Lessons Learned". *Proceedings of the 2000 Winter Simulation Conference*. eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick. pp. 855-858.

127. Michel, Rene and Martin Middendorf (1999) "An ACO Algorithm for the Shortest Common Supersequence Problem". *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England.

128. Middelkoop, T. and A. Deshmukh (1998) "Caution! Agent-based Systems in Operation". *International Conference of Complex Systems*.

129. Mihram, G. Arthur (1972) "Some Practical Aspects of the Verification and Validation of Simulation Models.". *Operations Research Quarterly*, vol 23., pp. 17-29.

130. *Modeling Human and Organizational Behavior: Application to Military Simulations*. Richard W. Pew and Anne S. Mavor, eds. National Academy Press: Washington DC. 1998.

131. Moore, Jacqueline and Richard Chapman "Application of Particle Swarm to Multiobjective Optimizations".

132. Morse, Philip M. and George E. Kimball (1998) *Methods of Operations Research*. Military Operation Research Society: Alexandria, Virginia. Reprinted in its entirety from © 1951 first edition printed by MIT Press and John Wiley & Sons, Inc.

133. Moscato, Pablo (1999) "Memetic Algorithms: A Short Introduction". *New Ideas in Optimization*. David Corne, Marco Dorigo, and Fred Glover, eds. The McGraw-Hill Companies: London, England.

134. Mullin, Tom (1991) "Turbulent Times for Fluids". *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

135. Murray, Carl (1991) "Is the Solar System Stable?". *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

136. National Geographic Society website. http://www.nationalgeographic.org

137. National Search and Rescue Committee (NSRC) (2000) United States National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual. http://www.uscg.mil/hq/g-o/g-opr/manuals.htm

138. Nayani, Nirupama and Mansooreh Mollaghasemi (1998) "Validation and Verification of the Simulation Model of a Photolithography Process in Semiconductor Manufacturing". *Proceedings of the 1998 Winter Simulation Conference*. eds. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan. pp. 1017-1022.

139. Naylor, Thomas H. and J. M. Finger (1967) "Verification of Computer Simulation Models". *Management Science*, vol. 14, no. 2, October. pp. B92-B104.

140. Page, Ernest H., Bradford S. Canova, and John A. Tufarolo (1997) "A Case Study of Verification, Validation, and Accreditation for Advanced Distributed Simulation". *ACM Transactions on Modeling and Computer Simulation*, vol. 7, No. 4. July. pp. 393-424.

141. Palmer, Tim (1991) "A weather eye on unpredictability". *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

142. Park, John, Nicholas Holford, and Bruce Charles (1999) "A Procedure for Generating Bootstrap Samples for the Validation of Nonlinear Mixed-Effects Population Models". *Computer Methods and Programs in Biomedicine*, vol. 59. pp. 19-29.

143. Parker, R. Gary and Ronald L. Rardin (1982) "An Overview of Complexity Theory in Discrete Optimizations: Part I. Concepts". *IIE Transactions*. vol. 14, no. 1. March 1982. pp. 3-9.

144. Parker, R. Gary and Ronald L. Rardin (1982) "An Overview of Complexity Theory in Discrete Optimizations: Part II. Results and Implications". *IIE Transactions*. vol. 14, no. 2. June 1982. pp. 83-89.

145. Peitgen, Heinz-Otto, Hartmut Jurgens, and Dietmar Saupe (1992) Chaos and Fractals: New Frontiers of Science. Springer-Verlag New York, Inc.: New York NY. 1992.

146. Pellegrini, Robert P. (1997) The Links Between Science, Philosophy, and Military Theory: Understanding the Past, Implications for the Future. Air University Press: Maxwell Air Force Base AL. August.

147. Percival, Ian (1991) "Chaos: a science for the real world". *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

148. Powell, Colin. Transcript of Remarks to the United Nations Security Council. New York NY. February 5, 2003. http://www.state.gov/secretary/rm/2003/17300.htm

149. Price, Joseph (2003) "Game Theory and U-Boats in the Bay of Biscay". Air Force Institute of Technology: Wright-Patterson Air Force Base OH. AFIT/GOR/ENS/03-18.

150. Reed website. http://homer.reed.edu/GkHist/Theropylae.html

151. Rieger, Larry and Emmet Beeker (2000) "Achieving Multi-Resolution Modeling Through Aggregation-Disaggregation". Phalanx. September 2000. pp. 8-9, 16.

152. Russell, Stuart J. and Peter Norvig (1995) *Artificial Intelligence: A Modern Approach*. Prentice Hall, Inc.: Upper Saddle River NJ.

153. Sanchez, Susan M. and Thomas W. Lucas (2002) "Exploring the World of Agent-Based Simulations: Simple Models, Complex Analyses". *Proceedings of the 2002 Winter Simulation Conference*. E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds. pp. 116-126.

154. Saperstein, Alvin (1995) "War and Chaos". American Scientist, vol. 84. November-December 1995. pp. 548-557.

155. Sargent, Robert G. (1991) "Simulation Model Verification and Validation". *Proceedings of the 1991 Winter Simulation Conference*. eds. Barry L. Nelson, W. David Kelton, and Gordon M. Clark. pp. 37-64.

156. Sargent, Robert G. (1996a) "Verifying and Validating Simulation Models". *Proceedings of the 1996 Winter Simulation Conference*. eds. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain.

157. Sargent, Robert G. (1996b) "Some Subjective Validation Methods Using Graphical Displays of Data". *Proceedings of the 1996 Winter Simulation Conference*. eds. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain. pp. 345-351.

158. Savit, Robert (1991) "Chaos on the Trading Floor". *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

159. Schlesinger, S., R. E. Crosbie, R. E. Gagne, G. S. Innis, C. S. Lalwani, J. Loch, R. J. Sylvester, R. O. Wright, N. Kheir, and D. Bartos (1979) "Terminology for Model Credibility". Simulation, vol. 32, no. 3. pp. 103-104.

160. Scott, Stephen (1991) "Clocks and Chaos in Chemistry". *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

161. Series, Caroline (1991) “Fractals, reflections and distortions”. *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY. Sheskin, David (2000) *Handbook of Parametric and Nonparametric Statistical Procedures, Second Edition*. Chapman & Hall/CRC: New York.

162. Shi, Jonathan Jingsheng (2001) “Practical Approaches for Validating a Construction Simulation”. *Proceedings of the 2001 Winter Simulation Conference*. eds. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer. pp. 1534-1540.

163. Shi, Y. and R. C. Eberhart (1999) “Empirical Study of Particle Swarm Optimization”. *Proceedings of the 1999 Congress on Evolutionary Computation*. July 6-9. pp. 1945-1950.

164. Silberschatz, Abraham and Galvin, Peter (1994) *Operating System Concepts*. Addison-Wesley Publishing Company: Reading MA.

165. Simpkins, Scott D., Eugene P. Paulo, , and Lyn R. Whitaker (2001) “Case Study in Modeling and Simulation Validation Methodology”. *Proceedings of the 2001 Winter Simulation Conference*. eds. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer. pp. 758-766.

166. Stephens, Cortez D. (2001) Decision Action Networks: Decision Making in Perception Based Models. *Maneuver Warfare Science 2001*. Gary Horne and Mary Leonardi, eds. Defense Automated Printing Service: Quantico VA.

167. Sycara, Katia (1998) “Multiagent Systems”. *AI magazine, vol. 19, No 2. Intelligent Agents*. Summer 1998.

168. Tighe, Thomas (1999) “Strategic Effects of Airpower and Complex Adaptive Agents: An Initial Investigation,” Air Force Institute of Technology: WPAFB OH. AFIT/GOA/ENS/99M-09.

169. Tighe, Tom, Ray Hill, and Greg McIntyre (2000) “A Decision for Strategic Effects: A Conceptual Approach to Effects Based Targeting”. *Aerospace Power Chronicles*. Air University: Maxwell Air Force Base AL.

170. Upton, Stephen (1998) “Warfare and Complexity Theory: A Primer”. *Maneuver Warfare Science 1998*. F. G. Hoffman and Gary Horne, eds. United States Marine Corps: Quantico VA.

171. “Verification and Validation: What Impact Should Project Size and Complexity Have on Attendant V&V Activities and Supporting Infrastructure”. Panel Presentation. Panel: James D. Arthur, Robert G. Sargent, James B. Dabney, Averill M. Law, and John D. (Jack) Morrison. Proceedings of the 1999 Winter Simulation Conference. eds. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans. pp. 148-155.

172. Vivaldi, Franco (1991) "An experiment with mathematics". *Exploring Chaos: A Guide to the New Science of Disorder*. Nina Hall, ed. W. W. Norton & Company, Inc.: New York NY.

173. Waddington, C. H. (1973) O.R. in World War 2: Operational Research against the U-Boat. Paul Elek (Scientific Books) Ltd.: London, England.

174. Warden, John III (1995) The Enemy as a System. *Aerospace Power Chronicles*. Maxwell Air Force Base AL. 1995.

175. Whealon, John (*imprimatur*) and Kenneth Shiner (*censor deputatus*) (1966, 1971, 1976, 1979) *Good News Bible: The Bible in Today's English Version. Catholic Study Edition.* William H Sadlier, Inc. and Thomas Nelson, Inc. New York NY.

176. White, T. and B. Pagurek (1998) "Towards Multi-Swarm Problem Solving in Networks". *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS '98)*. July. pp. 333-340.

177. Whitner, Richard B. and Osman Balci (1989) "Guidelines for Selecting and Using Simulation Model Verification Techniques". *Proceedings of the 1989 Winter Simulation Conference*. eds. E. A. MacNair, K. J. Musselman, and P. Heidelberger. pp. 559-568.

178. Widdowson, Brian (2001) "Finding and Exploiting (or Avoiding) the Nonlinearities Inherent in Warfare," *Maneuver Warfare Science 2001*, Gary Horne and Mary Leonardi, eds., Defense Automated Printing Service: Quantico VA.

179. Wirtz, James A. (1997) "A Joint Idea: An Antisubmarine Warfare Approach to Theater Missile Defense". *Airpower Journal*, vol. XI, no. 1. Air University: Montgomery AL. Spring 1997. pp. 86-95.

180. Wong, Raymond K. (1995) "Modeling and Simulation of Reactive Systems with Roles". *Proceedings of the 3rd Australian and New Zealand Conference on Intelligent Information Systems*. 27 November. pp. 1-6

181. Woodaman, Ronald (2000) "Agent-Based Simulation of Military Operations Other Than War Small Unit," Naval Postgraduate School: Monterey CA.

182. Woodcock, A. E. R., Loren Cobb, and John Dockery (1988) "Cellular Automata: A New Method for Battlefield Simulation," *Signal*, vol. 42, January, pp. 41-50.

183. Woods, Steve R. and Mario R. Barbacci (1999) *Architectural Evaluation of Collaborative Agent-Based Systems*. Technical Report CMU/SEI-99-TR-025. ESC-TR-99-025. October.

184. Wu, C. F. J. (1986) "Jacknife, Bootstrap and Other Resampling Methods in Regression Analysis". *Annals of Statistics*, vol. 14. pp. 1261-1295.

185. Young, Michael. Director, Air Force Research Laboratory, Human Effectiveness Directorate (AFRL/HES). Interview on August 8, 2001.

186. Zimm, Alan D. (2001) A Causal Model of Warfare. *Maneuver Warfare Science 2001*. Gary Horne and Mary Leonardi, eds. Defense Automated Printing Service: Quantico VA.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
12/19/03	Doctoral Dissertation	October 2000 - March 2004

4. TITLE AND SUBTITLE DEVELOPMENT APPROACHES COUPLED WITH VERIFICATION AND VALIDATION METHODOLOGIES FOR AGENT-BASED MISSION-LEVEL ANALYTICAL COMBAT SIMULATIONS	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Champagne, Lance E., Major, USAF	**5d. PROJECT NUMBER** DMSO 2002-058, AFRL/HES 2003-083 **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 641, WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/DS/ENS/03-02
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Modeling and Simulation Office DMSO Attn: Dr. Sue Numrich 1901 North Beauregard Street Alexandria, VA 22311 Air Force Research Laboratory AFRL/HES Attn: Dr. Michael Young Bldg 190, 2698 G St Wright-Patterson AFB, OH 45435	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT This research investigated the applicability of agent-based combat simulations to real-world combat operations. An agent-based simulation of the Allied offensive search for German U-Boats in the Bay of Biscay during World War II was constructed, extending the state-of-the-art in agent-based combat simulations, bridging the gap between the current level of agent-like combat simulations and the concept of agent-based simulations found in the broader literature. The proposed simulation advances agent-based combat simulations to "validateable" mission-level military operations.

Simulation validation is a complex task with numerous, diverse techniques available and levels of validation differing significantly among simulations and applications. This research presents a verification and validation taxonomy based face validity, empirical validity, and theoretical validity, extending the verification and validation knowledge-base to include techniques specific to agent-based models. The verification and validation techniques are demonstrated in a Bay of Biscay case study.

Validating combat operations pose particular problems due to the infrequency of real-world occurrences to serve as simulation validation cases; often just a single validation comparison can be made. This means comparisons to the underlying stochastic process are not possible without significant loss of statistical confidence. This research presents a statistical validation methodology based on resampling historical outcomes, which when coupled with the traditional nonparametric sign test, allows comparison between a simulation and historic operation providing an improved validation indicator than the single pass or fail test.

15. SUBJECT TER

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Raymond R. Hill (ENS)
U	U	U	UU	296	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4486; e-mail: Ray.Hill@wright.edu